

For Reference

NOT TO BE TAKEN FROM THIS ROOM

Ex LIBRIS
UNIVERSITATIS
ALBERTAENSIS



THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR Gregory J. Smith
TITLE OF THESIS Finite Element Pre-processor for
Hydrogeologic Modelling
DEGREE FOR WHICH THESIS WAS PRESENTED Master of Science
YEAR THIS DEGREE GRANTED Fall, 1984

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

THE UNIVERSITY OF ALBERTA

Finite Element Pre-processor for Hydrogeologic Modelling

by



Gregory J. Smith

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF Master of Science

The Department of Geology

EDMONTON, ALBERTA

Fall, 1984

THE UNIVERSITY OF ALBERTA
FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled Finite Element Pre-processor for Hydrogeologic Modelling submitted by Gregory J. Smith in partial fulfilment of the requirements for the degree of Master of Science.

ABSTRACT

An interactive computer package for generating a finite element mesh has been developed as a preprocessor for two and three-dimensional aquifer models. This scheme has the capability to generate triangular and quadrilateral elements in two-dimensions and tetrahedral and parallelepiped elements in three-dimensions. It is based on the Isoparametric mapping of quadrilaterals and can be used on flow domains that are described by both regular and irregular boundaries. The system is completely user interactive, utilizing a digitizing tablet and a colour graphics terminal to involve the user in all phases of the mesh generation, and also decrease the time and effort that is involved in more traditional methods of mesh construction. There are facilities to incorporate pre-positioned pumping wells and observation wells into the mesh, as well as constant head nodes, boundary fluxes and leakance elements. Output from the package is stored in a file in a format that is easily adaptable to most popular finite element codes. Some features of the code include: the ability to produce a hard copy of the mesh in any size within the limitations of the plotting hardware, and an optimization algorithm that renumbers nodes so as to increase computational efficiency. In addition to a description of the code, guidelines and hints are given for dealing with unusual shaped domains and for improving mesh design, and examples of the application of the package to a

variety of problems are presented.

ACKNOWLEDGEMENTS

The assistance provided by various people in the preparation of this thesis is gratefully acknowledged. Dr. F.W. Schwartz, the supervisor of this thesis and the person responsible for thinking up such practical research as this, provided much help throughout the writing of this work. I would like to thank the members of my examining committee , Dr. J. Toth, Dr. H.A.K. Charlesworth, Dr. D. Hackbarth and Dr. D.S. Chanasyk for their useful suggestions. Last, but not least, I would like to thank my wife, Laurie, for helping me through the past two years and supporting me to make all of this possible.

Table of Contents

Chapter	Page
1. INTRODUCTION	1
2. DESIGN AND IMPLEMENTATION OF A FINITE ELEMENT MESH GENERATOR	5
2.1 EXISTING METHODS	5
2.2 DESIGN OF A FINITE ELEMENT MESH GENERATOR FOR TWO- AND THREE-DIMENSIONAL PROBLEMS	8
2.3 DESCRIPTION OF THE PROGRAM	14
3. APPLICATIONS OF THE FINITE ELEMENT MESH GENERATION PROGRAM	24
3.1 TWO-DIMENSIONAL MESH GENERATION	24
3.2 THREE-DIMENSIONAL MESH GENERATION	34
4. OPTIMIZATION OF THE MESH GENERATION PROCESS	42
5. CONCLUSIONS AND RECOMMENDATIONS	51
REFERENCES	54
APPENDIX A User's Guide	57
APPENDIX B: Program Listing	65
APPENDIX C: Results of a Two-Dimensional Simulation	157

LIST OF TABLES

Table	Description	Page
1.	SUMMARY OF BASIC DATA FOR INDIVIDUAL LAYERS IN THE MODEL.....	34

LIST OF FIGURES

Figure		Page
1.	Flow domain illustrating designated corner points. Corner points are shown with an X.....	13
2.	Example 2-D problem.....	15
3.	Menu illustrating options available to a user to generate a mesh.....	16
4.	Flowchart illustrating the basic use of the program.....	18
5.	Computer generated 2-D mesh.....	28
6.	Hand produced 2-D mesh.....	29
7.	Map of area for 3-D simulation.....	31
8.	One layer computer generated 3-D mesh.....	33
9.	Drawdown cone after 15 years simulation period.....	35
10.	Figure demonstrating how the elements around a flux node are partitioned to give the impression of mesh compression. The solid lines represent the existing elements. The dashed lines represent the partitioning that is later smoothed.....	36
11a.	Basic rectangular shaped figure showing the domain discretized into 196 triangular elements.....	39
11b.	Basic rectangular shaped figure showing the domain discretized into 484 triangular elements.....	40

11c.	Basic rectangular shaped figure showing the domain discretized into 900 triangular elements.....	41
12.	Irregular shaped figure of a type to represent a buried channel.....	46
13.	Triangular shaped figure discretized with a mesh of triangular elements. Corner points used to generate the mesh are designated with a "X".....	47
14.	Triangular shaped figure showing results of designating corner points in a different manner. Corner points used to generate this mesh are designated with an "X".....	49

:

1. INTRODUCTION

Finite element methods for solving groundwater flow problems require the discretization of the flow domain, which involves the partitioning of the system under consideration into a grid or a mesh. Typically the mesh is constructed by hand through drawing a grid over the flow domain and manually entering the mesh parameters into a data set. This procedure, especially when one encounters more complex flow systems is very tedious, time consuming and subject to errors (Klienstreurer, 1980).

The need to automate this process is therefore apparent (Narasimhan and Witherspoon, 1980, and Kleinstreurer, 1980). Techniques based on point digitizers have proven useful in reducing the time for data entry of a manually prepared grid. However, there has been relatively little work in adapting fully automatic methods used for example in structural engineering to hydrogeologic problems. These schemes have been widely used, because they provide the ability to quickly and accurately change the density of the mesh to provide an optimal solution to a given problem.

In the fields of structural and automotive engineering the problem of automatic mesh generation appears somewhat less complex than for hydrogeological applications because of regular external and internal geometries. In hydrogeologic applications, mesh generation is complicated by the irregular geometry of the flow domain. Specifically one has to deal with:

- complex internal geometries, where it is necessary to model the boundaries of lithologic units;
- complex external geometries, as for example in the case of buried channels;
- providing specific locations for observation and pumping wells that must correspond to nodal locations; and
- the necessity of compressing the grid around a pumping well to increase the mathematical accuracy in an area where the hydraulic head may be changing quite markedly.

Such constraints on the mesh to be generated are usually not considered in the schemes presently available today. Existing techniques applied to groundwater modelling have been largely problem or program specific (Klienstreurer, 1980). These schemes generally involve fitting the flow domain to an idealized grid of regularly positioned elements. This kind of constraint usually makes it difficult to accurately model an irregular flow domain.

A more general procedure for mesh generation in a pre-processor form has been developed by Sartori et al. (1982). The meshes illustrated in their work appear to be accurate and well conditioned. This scheme fits a regularly positioned grid upon an irregular geometry such that elemental boundaries do not necessarily correspond to transmissivity boundaries.

Thus, there is still a need in hydrogeological applications to extend existing procedures for mesh

generation. The proven techniques in the field of structural engineering provide the logical starting point. Although many schemes exist for mesh generation, they are generally based on the following basic philosophies:

1. the mesh pattern is established by the computer from a minimal amount of information supplied in digital form; and
2. the positioning of the mesh is established by graphical-computer interaction using plotting tables or direct light pen displays (Zienkiewicz and Phillips, 1971).

The goal of my work is to extend these basic philosophies of mesh generation to hydrogeological problems. Specific objectives are: (1) to develop a mesh generator, capable of constructing a network of elements in two- or three-dimensions a level of complexity appropriate for most problems encountered in aquifer hydraulics, (2) to demonstrate the usefulness of this approach in solving practical problems, and (3) to demonstrate how grid design can improve finite element simulations.

The scheme for mesh generation used in this work is discussed in Chapter 2. It is based on an Isoparametric method developed by Zienkiewicz and Phillips (1971). The system is designed for maximum flexibility, user interaction, and is capable of generating triangular and quadrilateral shaped elements in a plane and projecting these elements in the third dimension to produce tetrahedral

and parallelepiped (brick type) elements. The mesh can be layered and hence can account for the patterns of geologic layering with relative ease.

The demonstration of the mesh generation scheme provided in Chapter 2 involves the simulation of hydraulic head distributions for both a two- and three-dimensional problem. The mesh for the two-dimensional case is prepared both manually and with the mesh generation program. The results of these two simulations are compared to verify the automatic procedure and to demonstrate its utility. The three-dimensional simulation is a more complex case designed to show how the mesh generation procedure can be applied to layered geological systems.

2. DESIGN AND IMPLEMENTATION OF A FINITE ELEMENT MESH GENERATOR

2.1 EXISTING METHODS

Techniques for automatic mesh generation have been in existence in their most basic form since the late 1960's. Thwacker (1980) provides an exhaustive review of these mesh generation techniques, including an outline of the basic theory, and an analysis of the drawbacks of each scheme. Current mesh generators employ one or a combination of two of the following schemes:

1. automatic triangulation;
2. coordinate transformations;
3. smoothing procedures; or
4. blending functions (Yerry and Shephard, 1983).

The automatic triangulation technique, a scheme demonstrated in the works of Cavendish (1974), Sadeck (1980), and Suhara and Fukuda (1972), allows the greatest flexibility in modelling a very general region. Cavendish (1974) uses a random number generator to fill the computational domain with nodes to some specified density. From this population of nodes, nodes are selected to form elements in a well conditioned mesh. Nodes that are not selected are extraneous and ignored.

Sadeck (1980) takes this approach a step further. Instead of randomly generating the coordinates of the nodes, the nodes are located on the basis of the position of

regularly spaced nodes around the boundary of the region in which the mesh is being constructed to form a well conditioned mesh. This procedure eliminates the need to generate the nodes first and then sort through them to produce the best conditioned element from the distribution of nodes.

The method used by Sadeck (1980) works well for regular geometric shapes, or simple forms with irregular geometry. For regions with complex geometry, which are discretized using quadrilateral elements, this method tends to form poorly conditioned elements especially where the boundary forms an acute angle. My experience with this method has shown that, although relatively simple to program it does require a substantial number of support subroutines to handle complex shapes. Yerry and Shephard (1983) state that although the triangulation algorithms have been developed that will work for general two-dimensional shapes, they have not proven popular because they occasionally produce ill-conditioned meshes and in many cases require substantial user interaction.

Coordinate transformations are used to map geometrically complex figures to a regular grid in the transformed domain. Having defined the transformation, the inverse transformation can be used to revert to the original domain. A disadvantage of this technique is that although the mesh is apparently well conditioned, depending on the coordinate transformation the mesh in reality may be very

distorted. As well, another disadvantage of this scheme is that the coordinate transformation must be provided before the computation can begin. This requirement is analagous to constructing an irregular grid (Thwacker, 1980).

Isoparametric methods (Zienkiewicz and Phillips, 1971) or, smoothing procedures as Yerry and Shephard (1983) describe them, involve the smooth distribution of nodes within the computational domain. The positioning of the interior nodes is determined by the mean of the coordinates of its neighbours. This condition is equivalent to an optimal grid, which corresponds to the equilibrium condition of a system of springs connecting the nodal points (Thwacker, 1980). This scheme requires that opposite boundaries of the domain have the same number of elements. This constraint is a disadvantage because the user can not easily change the density of the elements within the domain.

2.2 DESIGN OF A FINITE ELEMENT MESH GENERATOR FOR TWO- AND THREE-DIMENSIONAL PROBLEMS

My design for an automatic mesh generator for three-dimensional groundwater flow problems is based upon the work of Zienkiewicz and Phillips (1971), which involves the application of isoparametric coordinates, or smoothing procedures as Yerry and Shephard (1983) had described them in the previous section. The work by Zienkiewicz and Phillips (1971) deals mainly with two-dimensional meshes, and indicates that the method can be easily adapted to three-dimensional problems.

The code that I have developed has the capability of generating finite element meshes in both two- and three-dimensions. It discretizes a domain using either quadrilateral or triangular elements in two-dimensions, and parallelepiped or tetrahedral elements in three-dimensions.

To facilitate both the rapid and simple entry of data into the code, a menu oriented approach has been implemented using a Summagraphics Corp. ID series tablet digitizer to input the bulk of the digital information. This approach greatly simplifies digital input of coordinate information, and provides the user with flexibility in controlling the pattern of mesh generation.

Coupled to the digitizer is an Advanced Electronics Design AED512 colour graphics terminal that completely supports the graphics packages necessary to display and manipulate the networks that are generated. The Integrated

Graphics (Davis, 1977) system is a library of FORTRAN-callable subroutines for performing graphics operations. These systems are all supported by the Amdahl 5860 mainframe at the University of Alberta.

With this hardware, the amount of data that must be input manually can be greatly minimized to generate a mesh consistent with the following criteria:

1. the geometry of the boundaries, whether singly or multiply connected, must be accurately described;
2. a facility for describing the spatial variability in the hydraulic properties of the geologic material, wherever these occur;
3. a facility for grading the mesh so as to achieve different kinds of mesh compression in specific sub-areas of the domain;
4. an achievement of shapes which are approximately equidimensional and do not lead to numerical ill-conditioning;
5. a renumbering system which results in the best computational efficiency (Zienkiewicz and Phillips, 1971; Collins, 1973).

The boundary of the domain or sub-domain in any plan view can be adequately described by digitizing points around this boundary. Zones of differing transmissivities are digitized separately, with the program establishing the required continuity in the mesh geometry along boundaries. The manner in which the other three criteria are implemented

in the procedure for mesh generation will become apparent in the following sections.

The mesh generating scheme is based on the isoparametric mapping of quadrilaterals. For any shaped domain the coordinates of the interior nodes can be described by the following equations (Zienkiewicz and Phillips, 1971):

$$\begin{aligned}x &= N_i x_i \\y &= N_i y_i \\z &= N_i z_i\end{aligned}\tag{1}$$

where: N_i = the shape factor determining the placement the nodes around the domain boundary.

x_i = x-coordinate of a regularly spaced node being mapped onto the computational domain.

y_i = y-coordinate of a regularly spaced node being mapped onto the computational domain.

z_i = z-coordinate of a regularly spaced node being mapped onto the computational domain.

The shape factor N_i is described in terms of a coordinate transformation based upon the geometry of the domain boundary. In the field of hydrogeology, the geometry of the flow domain is not easily described as a simple equation.

The approach taken in this thesis involves defining the boundaries of the figure by a series of regularly spaced nodes. The spacing of these nodes is based on the desired distribution of elements within the figure. In three-dimensions the nodes are regularly spaced along the twelve edges or folds of the figure.

To obtain a regular spacing along an irregular boundary, a method based on Lagrange polynomials (Hornbeck, 1975) is incorporated into the code. Essentially this Lagrangian technique produces polynomials that are "tied down" (Hornbeck 1975) at the data points given in the digitization and interpolates positions in between. In the code, boundaries of the domain or sub-domain are represented by equations of up to a third order polynomial, which can represent most geometries that may be encountered.

The Lagrangian method interpolates evenly spaced points from an irregular distribution of sample points. For this method, consider a series of points described by a function $f(x_i)$ where the x_i are in general not evenly spaced and i can take on integer values 0 to n (meaning there are $n+1$ data points). To fit a polynomial interpolation between the x_i 's is to fit a polynomial of degree n to these $n+1$ points. The mechanics of this method in this code is a moving average approach through the data points as a constantly changing polynomial. That is, utilizing four data points at a time, the code calculates evenly spaced points within a range represented by these points. As the point generation

progresses, the sample points change to keep the generated points within the range represented by the four data points.

Because this scheme is based on the mapping of quadrilaterals, it is necessary to designate four corner points (in three-dimensions, eight corner points) of the irregular figure to obtain an equal number of regularly spaced nodes on opposite sides of the figure. With the digitized figure displayed on the graphics terminal, the user, by positioning the cursor defines the four (or eight) corners of the figure. In dealing with irregular geometries, corner points are not immediately obvious but must be designated in a manner to provide an even distribution of elements as illustrated in Figure 1.

With the boundary nodes evenly spaced around the boundary of the domain under consideration, the boundaries are completely defined but not in the equation form as in the work of Zienkiewicz and Phillips (1971). The distribution of the interior nodes becomes based on the distribution of the boundary nodes and may be described by the following set of equations:

$$N_x = (x_1 - x_2) / NELM_x$$

$$N_y = (y_1 - y_2) / NELM_y$$

$$N_z = (z_1 - z_2) / NELM_z$$

(2)

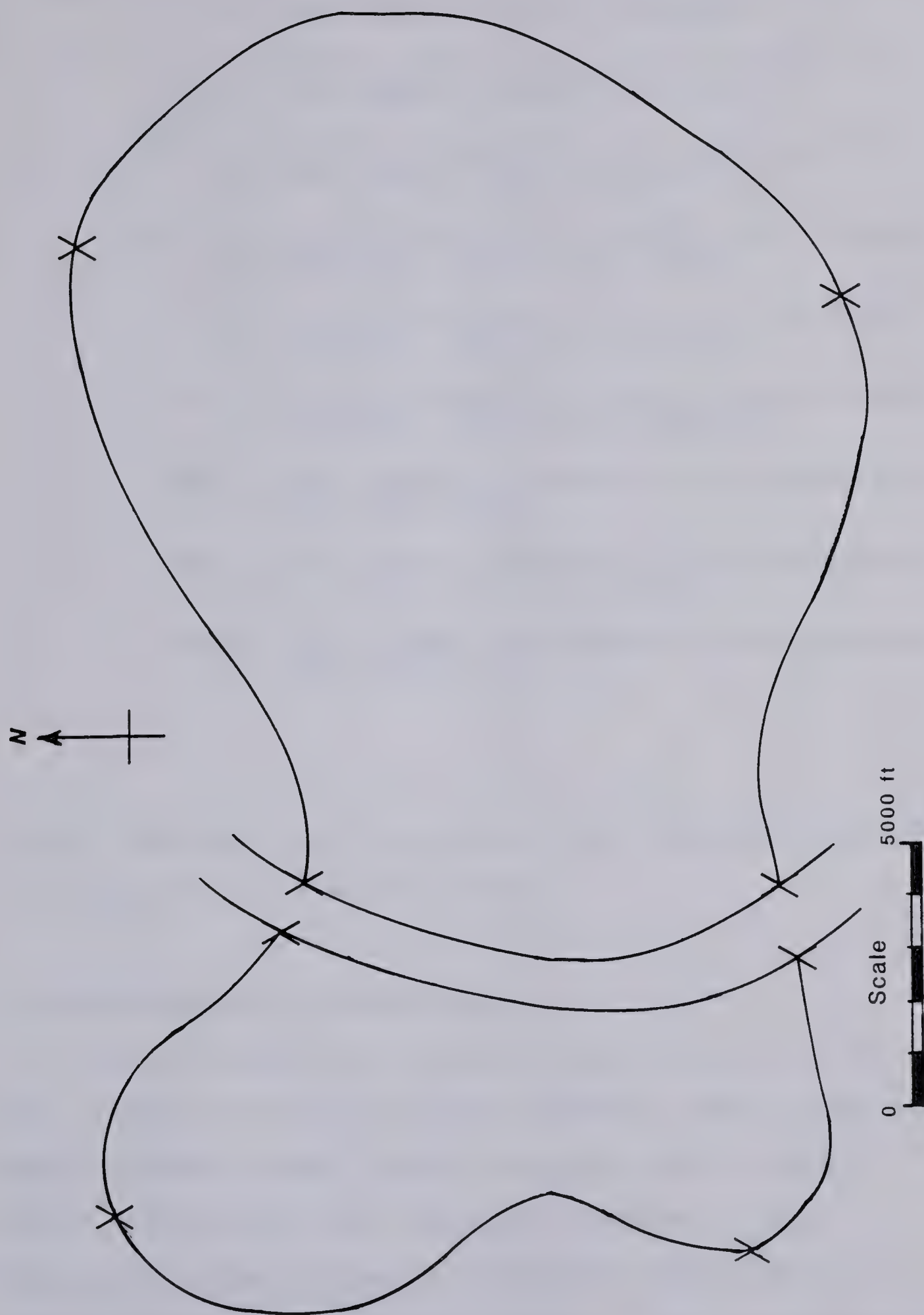


Figure 1. Flow Domain illustrating designated corner points. Corner points are shown with an X.

where: N_x = the factor determining the distribution of the nodes in the x-direction.

N_y = the factor determining the distribution of the nodes in the y-direction.

N_z = the factor determining the distribution of the nodes in the z-direction.

x_1 & x_2 = corresponding x-coordinates of nodes on opposite sides of the domain.

y_1 & y_2 = corresponding y-coordinates of nodes on opposite sides of the domain.

z_1 & z_2 = corresponding z-coordinates of nodes on opposite sides of the domain.

$NELM_x$ = the number of elements to be generated in the x-direction.

$NELM_y$ = the number of elements to be generated generated in the y-direction.

$NELM_z$ = the number of elements to be generated in the z-direction.

As the nodes are being generated, the elemental incidences are being described simultaneously.

2.3 DESCRIPTION OF THE PROGRAM

The mesh generation program consists of 13 parts: the main program and 12 subroutines: SMOOTH, DSTRB, NODE, ELMNT, SETUP, OPTNUM, NDEGEN, INTANG, ASKPLT, CNCTQ, CNCTT1, CNCTT2, as well as the Integrated Graphics Library Subroutines. The program is presently dimensioned to generate 5000 nodes and 3000 elements, which should be more than sufficient for most problems.

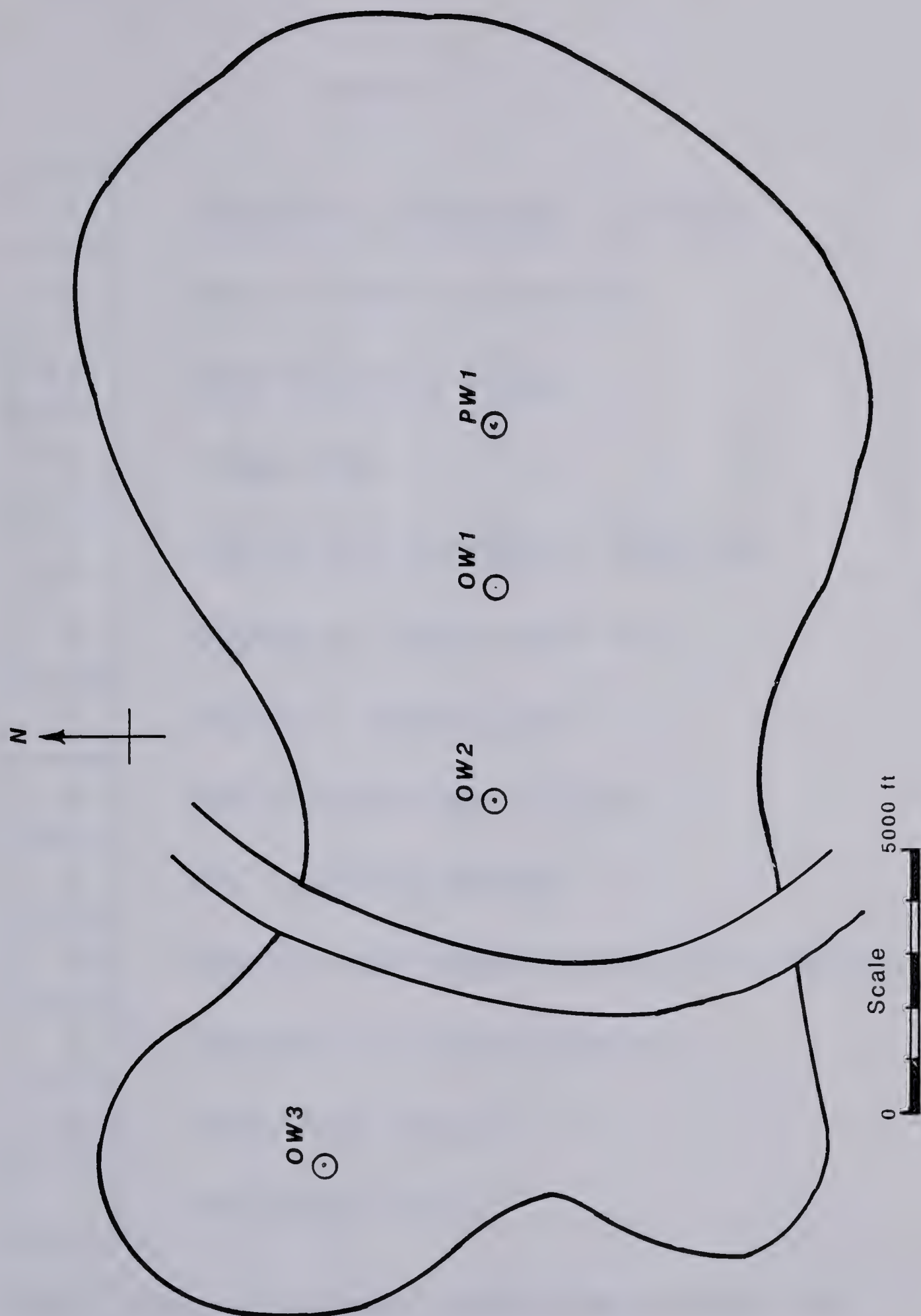


Figure 2. Example 2-D problem.

MENU

13	Generate a Mesh(after digitizing)
12	Output Data for Later Use
11	Input Boundary Fluxes
10	Future Use
9	Digitize a Transmissivity Boundary
8	Digitize an Observation Well
7	Digitize a Pumping Well
6	"Pic" Constant Head Nodes
5	"Pic" Leakance element
4	Remove Last Transmissivity Zone Generated
3	Remove Last Point Digitized
2	Stop/Begin Again
1	Re-display Menu

Figure 3. Menu illustrating options available to a user to generate a mesh.

The MAIN program initializes all the indices used in the entire program, reads in the title of the simulation and interactively sets the basic options of the program. These options determine: whether one is working in English or metric units; the scale; whether it is to be a two-dimensional or three-dimensional simulation; type of elements required; and if in three-dimensions whether a steady state or transient simulation is to be performed. The MAIN program also contains the Menu from which any of the other subroutines will be called. This is implemented as a series of 13 switches that may be accessed at any time during the construction of the mesh in a manner outlined in the flow chart, Figure 4. The 13 switches allow the user to perform a variety of tasks to synthesize a mesh.

Switch No. 1 simply re-displays the menu. This feature was incorporated into the design of the program to provide the user with an easy way to recall all of the options. Carrying most of the operating instructions with the program eliminates the need for a paper copy of the menu that could be misplaced.

Switch No. 2 activates a built-in procedure to restart or terminate the program. The instructions outlined in the user's guide in Appendix A explains how to start the program. However because the user has options throughout the entire program that may never be used, an on/off switch is provided to terminate the program.

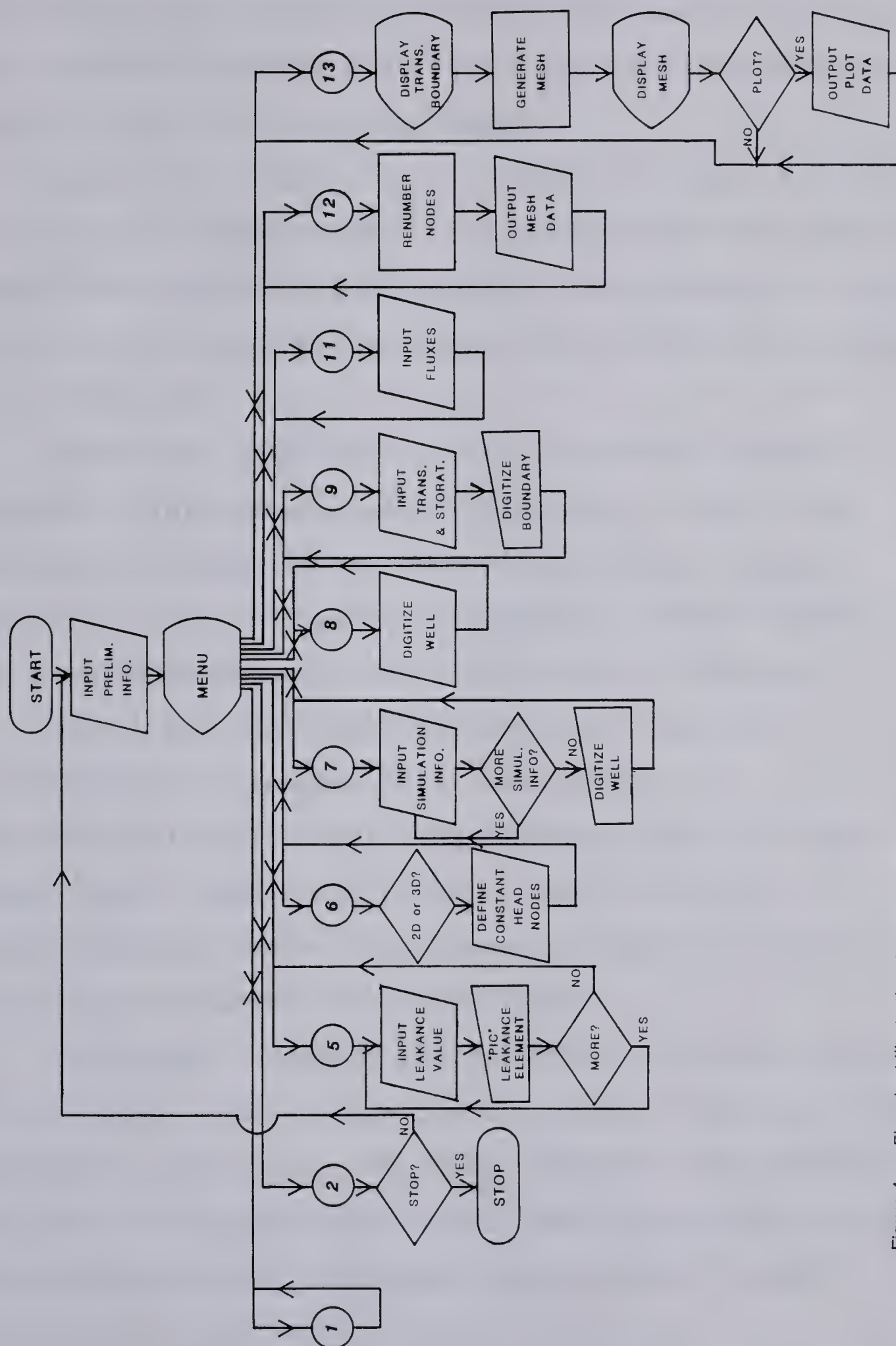


Figure 4. Flowchart illustrating the basic use of the program.

Switch No. 3 deletes the previously digitized point. This feature is incorporated so that one could take care of such hardware problems such as a faulty digitizer switch, static on the line and human error.

Switch No. 4 allows one to remove the mesh generated for the last transmissivity zone incorporated into the main mesh. This feature avoids repeating large amounts of data entry in the case where a mistake in the digitizing process goes unnoticed.

Switch No. 5 allows the user to define leakance elements, (ie: recharge associated with a river). This feature only works in two-dimensions, using a graphic routine to pick out particular elements ("Pic" routine) with the program prompting a numerical value of leakance.

Switch No. 6 defines constant head nodes in two-dimensions through a "Pic" routine and in three-dimensions constant head boundary faces. The user cannot define individual constant head nodes in three-dimensions with this program because the "Pic" routine cannot easily define the z-coordinate.

Switch No. 7 defines the location of a pumping well in the x-y plane. As well the program prompts the user to input information relating to the pump schedule. The position of the well is incorporated into the mesh as a node and in three-dimensions is continuous throughout all layers subsequent to its initial nodal definition.

Switch No. 8 defines the location of an observation well in a similiar manner to the pumping well.

Switch No. 9 allows the user to define the boundaries of a transmissivity zone with the program prompting for values of storativity and transmissivity (in three-dimensions, elemental hydraulic conductivities in x-, y- and z-directions).

Switch No. 10 is designated for future use. It was originally used to renumber the nodes to reduce bandwidth of the mesh, but this feature was subsequently incorporated into Switch No. 12.

Switch No. 11 designates boundary fluxes across the entire face of a mesh. It was felt that a boundary flux feature was needed in the mesh generation scheme, but to "Pic" out individual elements would be too costly and time consuming.

Switch No. 12 renumbers the nodes of the mesh to reduce bandwidth (Collins, 1973) and outputs the information in a format that is compatible with the Verge-Frind (Verge, 1972) two-dimensional code dealing with transient simulations for confined aquifers. In three-dimensions, a format compatible with Verge's (1975) three-dimensional groundwater flow model with minor changes in the input. The numerical information is output to device 7 and the plotting information to device 9. The workings of Collins' (1973) renumbering scheme will be discussed in Section 4.

Switch No. 13 activates the mesh generation procedure. This procedure is interactive with the user, allowing him to define the configuration of the mesh and the elemental density. If any pumping wells are present, the program will concentrate elements around the well and smooth the distribution before it is displayed on the terminal. Having displayed the mesh on the terminal, the user can rotate it to better visualize its faces and plot it on a hardcopy plotter.

The individual subroutines used in the code correspond to no particular switch discussed previously. A switch may have as many as five subroutines associated with it or it may have none. For this reason, the function of each subroutine will be discussed briefly.

Subroutine SMOOTH, used in conjunction with subroutines CNCTQ, or CNCTT1, or CNCTT2, concentrates elements around pumping wells and smooths the distribution of elements in the vicinity of the wells. Subroutine CNCTQ applies to quadrilateral and "brick type" elements around wells, whereas CNCTT1 and CNCTT2 apply to triangular and tetrahedral elements.

The concentrating subroutines CNCTQ, CNCTT1, or CNCTT2 partition the previously generated elements around a pumping well into smaller elements. The smaller elements are numbered and the incidences defined. The incidences of the elements around these partitioned elements also have their incidences redefined. A large part of these subroutines can

be thought of as being largely devoted to bookkeeping duties in the storage of elemental incidences.

Subroutine SMOOTH takes the coordinates of the newly generated nodes in CNCTQ, CNCTT1, or CNCTT2 and repositions them using an averaging technique based on the position of the surrounding nodes. This provides for a smooth distribution of elements in the vicinity of the pumping well.

Subroutine DSTRB re-distributes the digitized points around a perimeter of the transmissivity zone using the Lagrange interpolation routine (Hornbeck, 1975)

Subroutine NODE uses the redistributed nodes' positions to produce nodes within the interior of the transmissivity zone, with the elemental incidences being described by calls to subroutine ELMNT.

Subroutine SETUP and OPTNUM are based on algorithms developed by Collins (1973) and modified for use in this program. SETUP, as the name implies, sets up arrays that describe the interconnection relationship of each node. The purpose of these arrays is to make it easier for the renumbering routine to know which nodes are interconnected, hence the differences in numbering these nodes should be small. OPTNUM takes the results from SETUP and renumbers the nodes in the mesh based on each node's connection with its neighbours so that the bandwidth is reduced.

Subroutines INTANG and NDEGEN are used in conjunction with the subroutines for mesh compression. INTANG calculates

the interior angle formed by the positioning of three specified nodes. NDEGEN uses the angle determined in INTANG and a specified radial distance defines a node in rectangular coordinates from polar coordinates.

Subroutine ASKPLT provides the user with the capability to generate a hardcopy plot. ASKPLT also allows the user to change the size of the plot. The default size is the same size that is viewed on the screen.

3. APPLICATIONS OF THE FINITE ELEMENT MESH GENERATION PROGRAM

3.1 TWO-DIMENSIONAL MESH GENERATION

To illustrate the operation and ease of use of this program, the following sample problem is included. Figure 2 depicts the two-dimensional domain of interest. The groundwater system consists of a confined aquifer enclosed on all sides by no flow boundaries, with a river providing a source for induced infiltration. There is one pumping well and three observation wells as shown on Figure 2. The transmissivity of the aquifer east of the river is 4.0×10^{-4} m²/s, while west of the river it is 2.0×10^{-4} m²/s. The pumping well is assumed to be pumped at a constant rate of 6.32×10^{-4} m³/s. The leakance rate from the river is 1.0×10^{-8} m/s. This is the basic information that is required to produce a mesh.

Once execution of the program has begun using the run command in Appendix A, the first prompt will be to enter: 1) a title; 2) the units one is working in, (ie: English or metric); 3) the scale of the figure from which one will be producing the mesh; 4) whether one is working in two- or three-dimensions; 5) and whether one desires quadrilateral (brick type) or triangular (tetrahedral) type elements.

Next, the user is requested to clear the digitizer, and digitize the lower left, lower right, and upper left corners of the page of the figure that is to be digitized. The

purpose of this step is to scale the figure for display on the graphics terminal.

The menu is now displayed on the screen and the user follows the steps outlined in the flow chart, Figure 4.

The first step is to digitize the boundaries of a given transmissivity zone, and fixing the transmissivity value of all the nodes in this zone. Once the transmissivity boundary is defined, the user would then re-display the menu by digitizing Switch No. 1. Before moving to the next transmissivity zone, it is necessary to define all observation and pumping wells. For the problem depicted in Figure 2, the first transmissivity zone contains two observation wells and one pumping well. By selecting Switch No. 7, the user can enter the pumping information (ie: start and finish times of the simulation, number of time steps and pumping rates) as requested, and digitize the pumping well location. Observation well locations are input using Switch No. 8. With this information, the user is able to generate a mesh for this transmissivity zone by selecting Switch No. 13. The outline of the transmissivity zone is displayed onto the screen. Because this scheme is based on the mapping of quadrilaterals, the user is prompted to indicate with the cursor the four corners of the figure (even if one desires triangular elements).

Once the four corners are indicated, the program generates boundary nodes using a method of interpolation through Lagrange polynomials (Hornbeck, 1975). The prompts

request the four corners of the figure, (see Figure 1), and request the number of elements in both the x and y directions.

An important feature of this program is that observation and pumping wells are digitized prior to the mesh generation so that nodes can be placed at these locations and incorporated in the mesh.

If the transmissivity zone contains any pumping wells, the mesh is smoothed. This smoothing routine adjusts the positions of each of the interior nodes to make sure that each of the elements are well conditioned. Also, elements are concentrated around pumping wells (see Section 4) to mathematically handle the large changes in hydraulic head encountered here. The resultant mesh is then displayed on the screen with the query "PLOT?". Because we have not completed our mesh generation process the response would be , "no", and the menu would then be redisplayed on the screen.

The next area of interest would be the river because the boundaries of elements must correspond to the boundaries of the river for the purposes of defining leakance elements. Selecting Switch No. 9 again, we enter the boundaries of the river as a separate transmissivity zone, and generate a mesh as before. Leakance elements are defined by selecting Switch No. 5, which the user does by indicating, with the cursor, the leakance elements and the value of leakance. This routine can be rather tedious, but does allow one to define leakance elements anywhere on the mesh.

The user could continue the mesh generation process with the area west of the river.

With the mesh completely drawn, the user can choose to have it plotted by sending the plot data into a file on output device 9 for the hard copy plotter.

By selecting Switch No. 12, the nodal points are renumbered to reduce bandwidth (Collins, 1973), and output into a file attached to device 7 in a format that is compatible with the Verge-Frind (Verge, 1972) two-dimensional, finite element program for confined aquifer simulations.

The mesh generated using this procedure is shown in Figure 5. It can be compared with the hand-drawn version illustrated in Figure 6. The hand-drawn mesh has 104 nodes, 177 elements, with a bandwidth of 23, whereas the computer generated mesh has 160 nodes, 272 elements with a bandwidth of 21. The hand drawn version required approximately 6 hours to draw, number, enter the data file and check for errors. The computer generated mesh required approximately 6 seconds of processing time on an Amdahl 5860 mainframe to produce, and approximately 20 minutes at the terminal to enter the necessary data.

The simulated hydraulic head distribution due to seven days of pumping, based on the computer generated mesh, is presented in Appendix C. By comparing these results with those simulated using the hand-generated mesh, it is possible to see that the results are very similiar. The

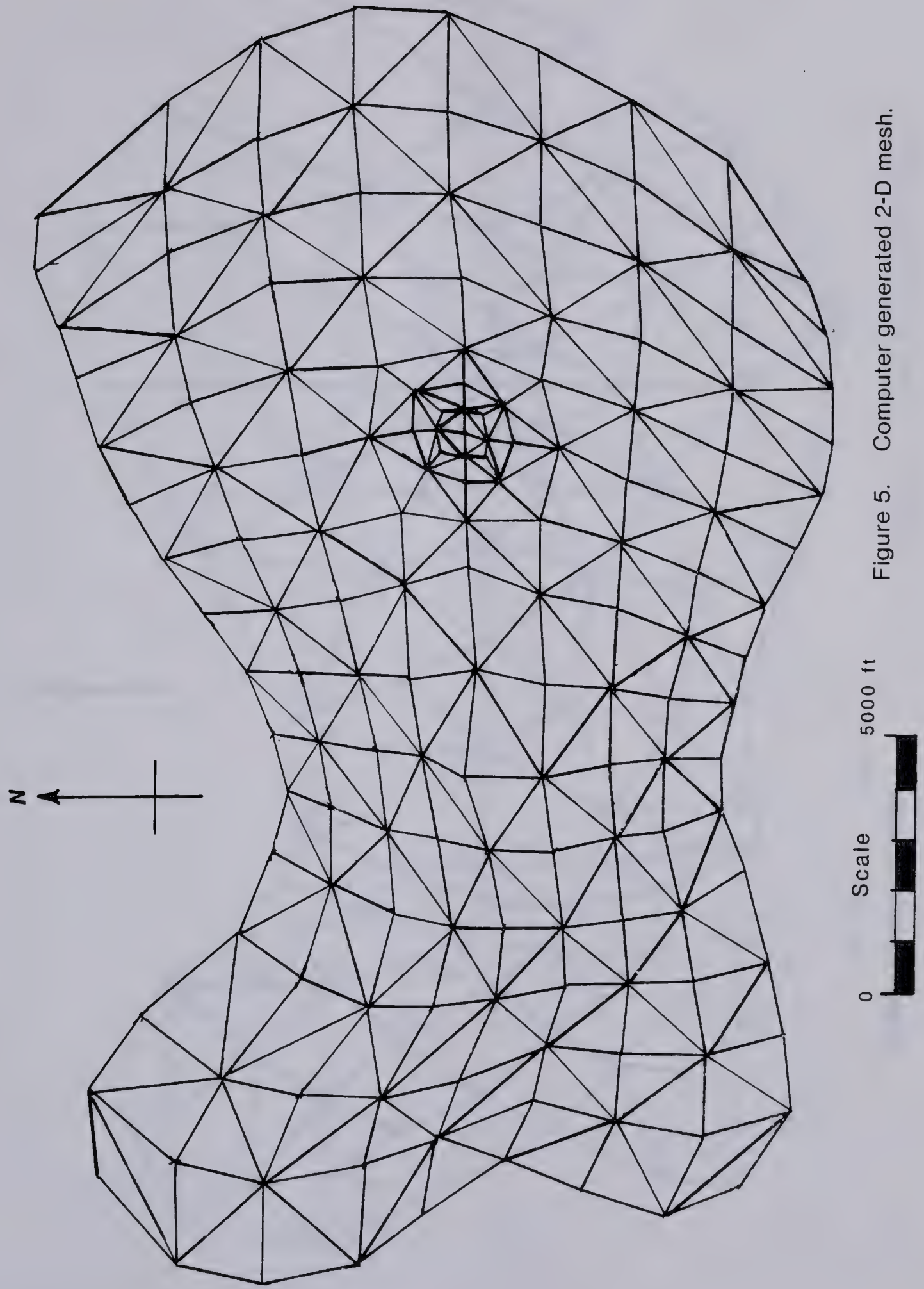


Figure 5. Computer generated 2-D mesh.

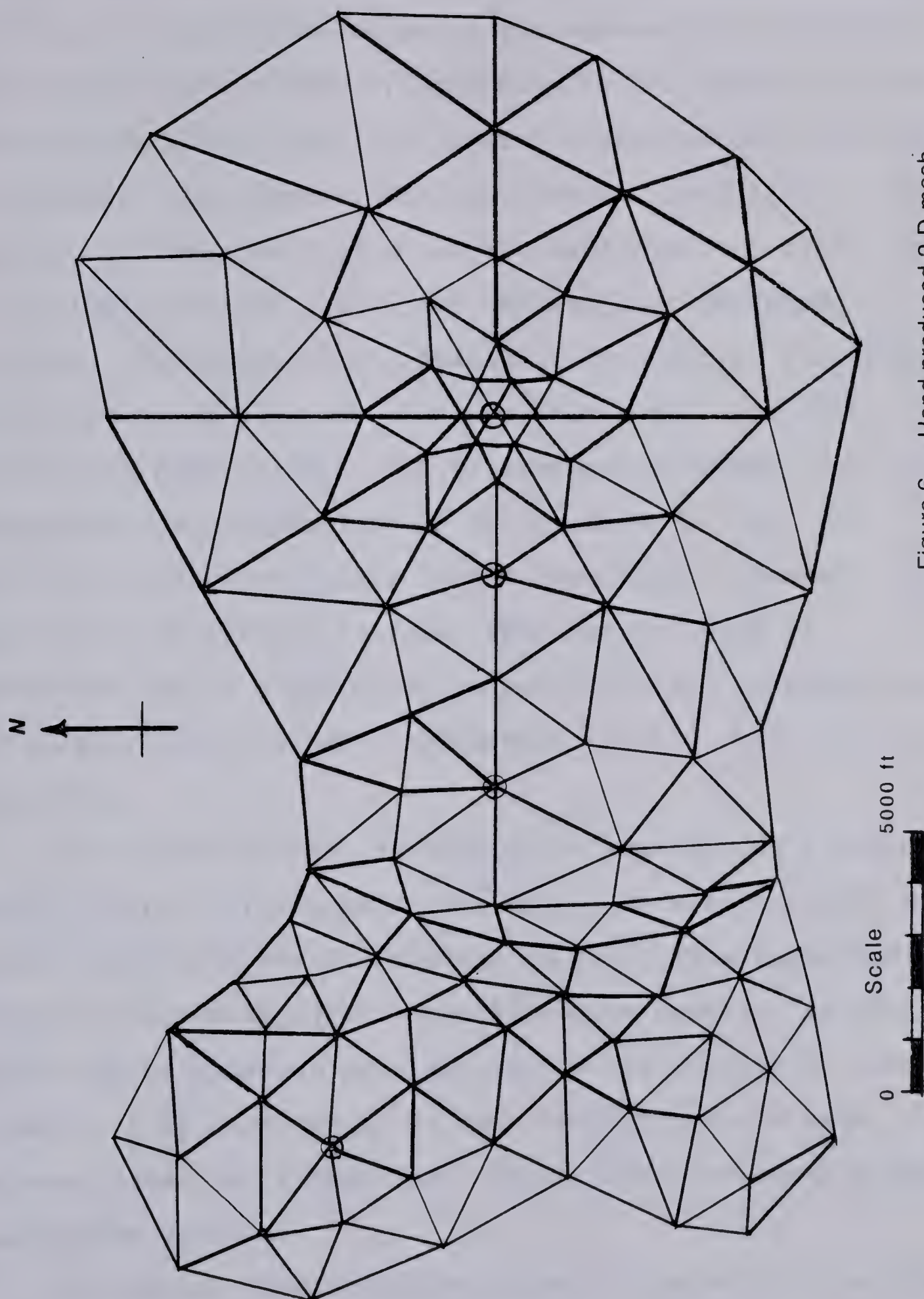


Figure 6. Hand produced 2-D mesh.

original purpose of this exercise was to determine whether it would be worthwhile to go to the expense of installing observation well number 3. According to the results of the simulations we can see that in both instances one would have to question the expense, for the drawdown indicated at this point would only be 0.03 m for the hand-drawn version, with a similar value of 0.02 m for the computer generated version. The values of the hydraulic head differ from less than 0.01 ft in the area of observation well 3 to 3 ft. around the pumping well. The discrepancies between the two simulations are approximately in the order of 10%. The source of this discrepancy is not immediately apparent, but may be due to several sources. The sources could be accounted for in differences in mesh density, computational efficiency, and inherent inaccuracies within the simulation algorithm.

Variations in mesh density could account for a certain amount of this discrepancy. The higher density may have been better able to cover the changes in flux. This feature will be more fully discussed in the following chapter. As well, there may be inherent problems within the simulation program itself. In my experience, as much variation as is seen between these two simulations can be seen when viewing many hand drawn versions.

To further illustrate the capabilities of my algorithm, additional two-dimensional examples are included in this section. Figures 11a, 11b and 11c illustrate how easy it is,

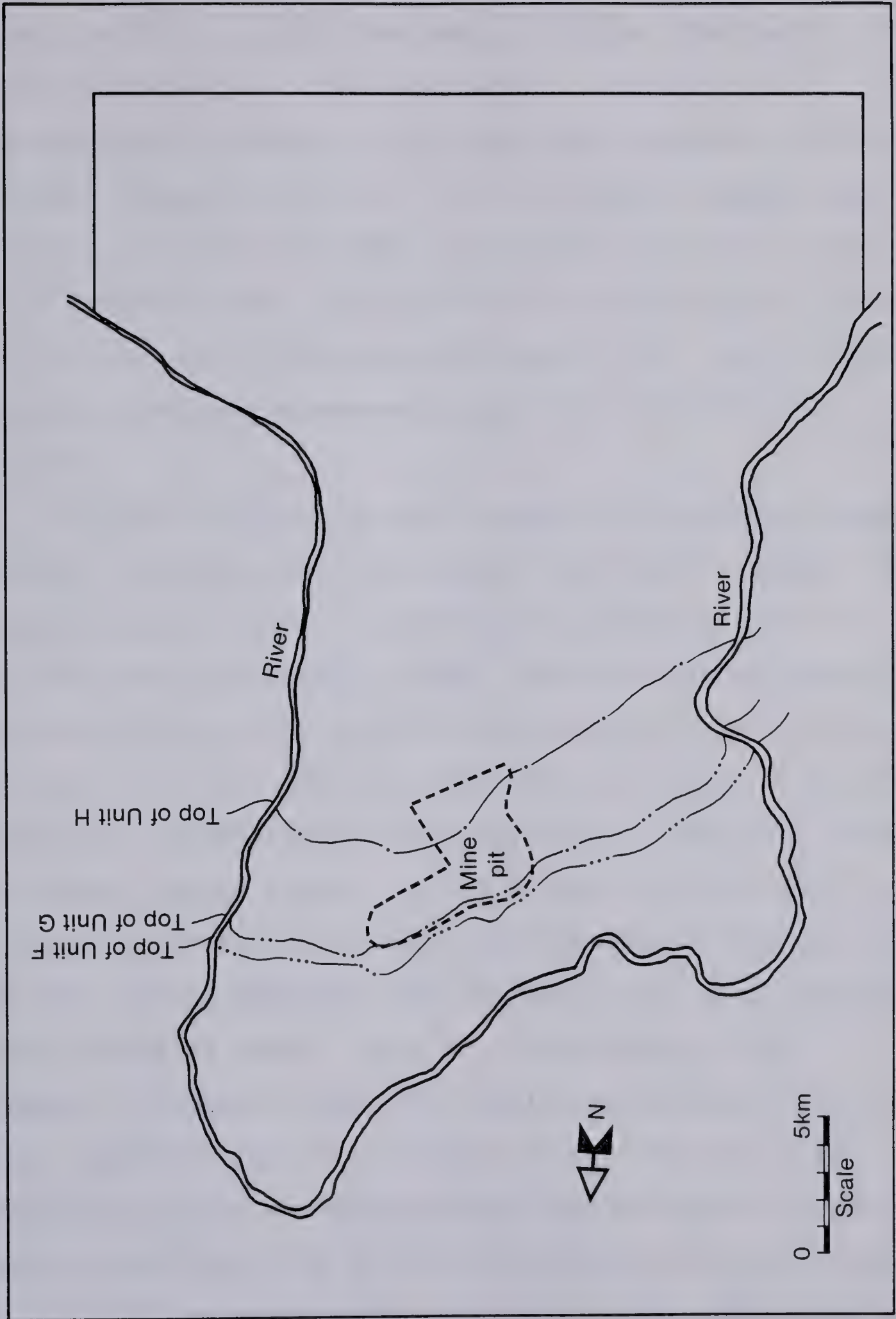


Figure 7. Map of area for 3-D simulation

with an automatic mesh generator, to increase the nodal density within a single rectangular domain. The facility to alter the density of the nodes within a domain is represented by a prompt within the mesh generation routine accessed through Switch No. 13. This prompt requests that the user indicate the number of elements desired in the x and y directions and, if working in three-dimensions, the z-direction. It is therefore the duty of the user to insure that the resultant elements are as well conditioned as possible.

Figures 8 through 14 are included to show what meshes actually look like when the domains take on a somewhat more complex shape. Figure 13 represents a shape that may be encountered if one were to model a buried channel situation. The four corners were placed as shown with an "X". Figures 13 and 14 show how important the initial placement of four corners is in controlling the character of the grid. With a triangular shaped region, it can be seen that the mesh in Figure 14 represents a smooth distribution of elements from the apex to the opposite side. In Figure 14, with the corner nodes placed as shown, there is a stretching of the elements, producing some ill-conditioned elements at the lower boundary. This last figure illustrates how it is possible to alter the mesh density along a given boundary by altering what would be an obvious placement of corner nodes to one that would alter the distribution of nodes along a given boundary. In this way any adjacent zones would not be

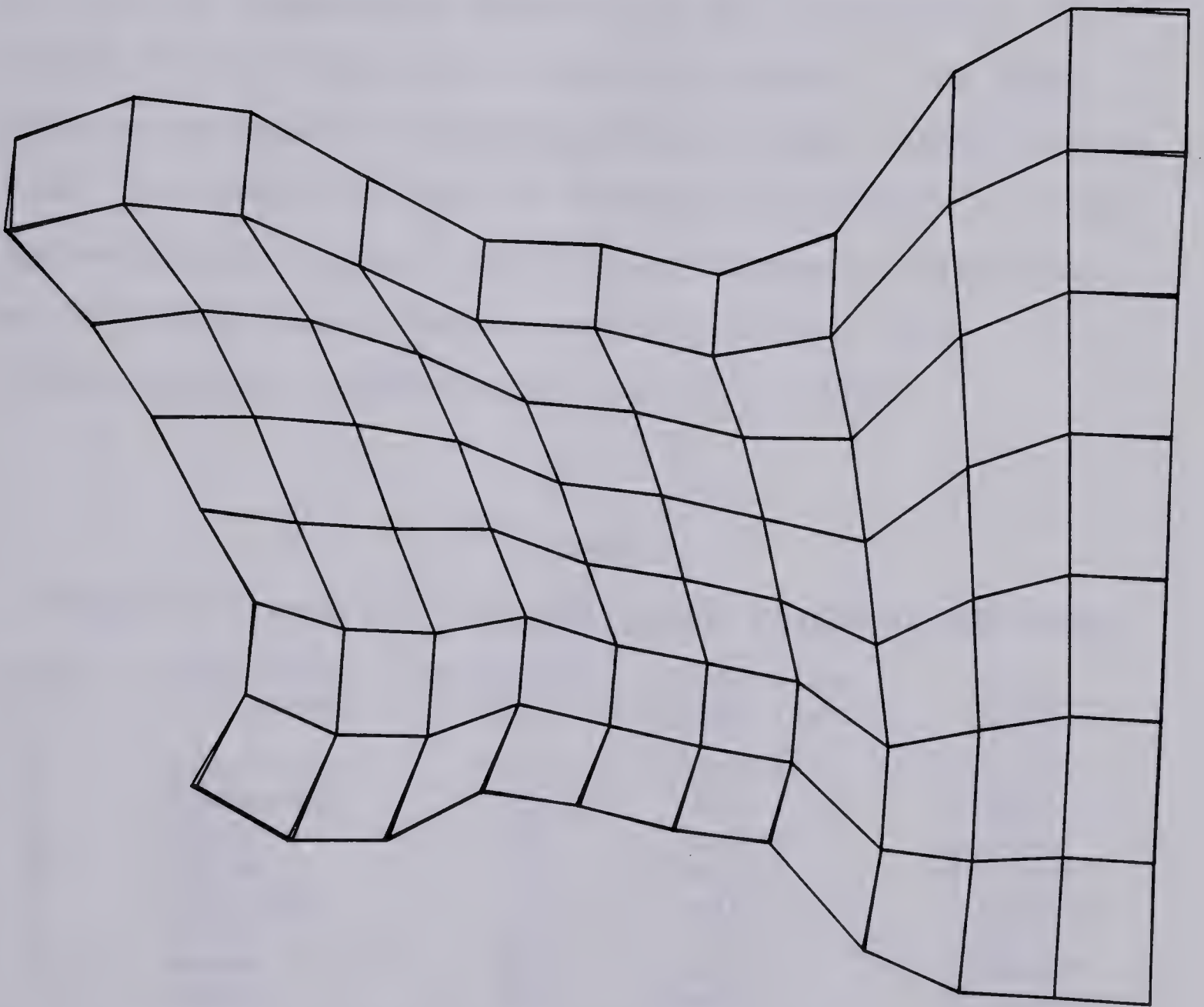


Figure 8. One layer computer generated 3-D mesh.

constrained to a set mesh density.

3.2 THREE-DIMENSIONAL MESH GENERATION

To illustrate the capabilities of the code in three-dimensional applications, consider the following hypothetical problem of assessing the influence of an open pit mine on groundwater levels. The map illustrating the layout of the simulation is shown in Figure 7. For this problem the system is discretized into eight transmissivity zones representing a layered sedimentary sequence striking approximately east-west and dipping southward. Layers E, G and H are assumed to outcrop at the surface. The hydrogeological parameters are outlined in Table 1.

TABLE 1.

SUMMARY OF BASIC DATA FOR INDIVIDUAL LAYERS IN THE MODEL

Layer No.	Predominant Lithology	Thickness (m)	Hydraulic Cond. (m/s)	Trans. (m^2/s)
H	limestone	125	1×10^{-6}	1.25×10^{-4}
G	limestone	30	1×10^{-7}	3.0×10^{-6}
F	shale	20	1×10^{-8}	2.0×10^{-7}
E	shale	15	5×10^{-7}	7.5×10^{-6}
D	ore zone	15	1×10^{-6}	1.5×10^{-4}
C	shale	20	1×10^{-7}	2.0×10^{-6}
B	shale	30	1×10^{-9}	3.0×10^{-8}
A	shale	50	5×10^{-8}	2.5×10^{-6}

The storativity is assumed to be constant throughout at a value of 0.0001.

For the purposes of defining the flow domain, the rivers bounding the three sides of the system are

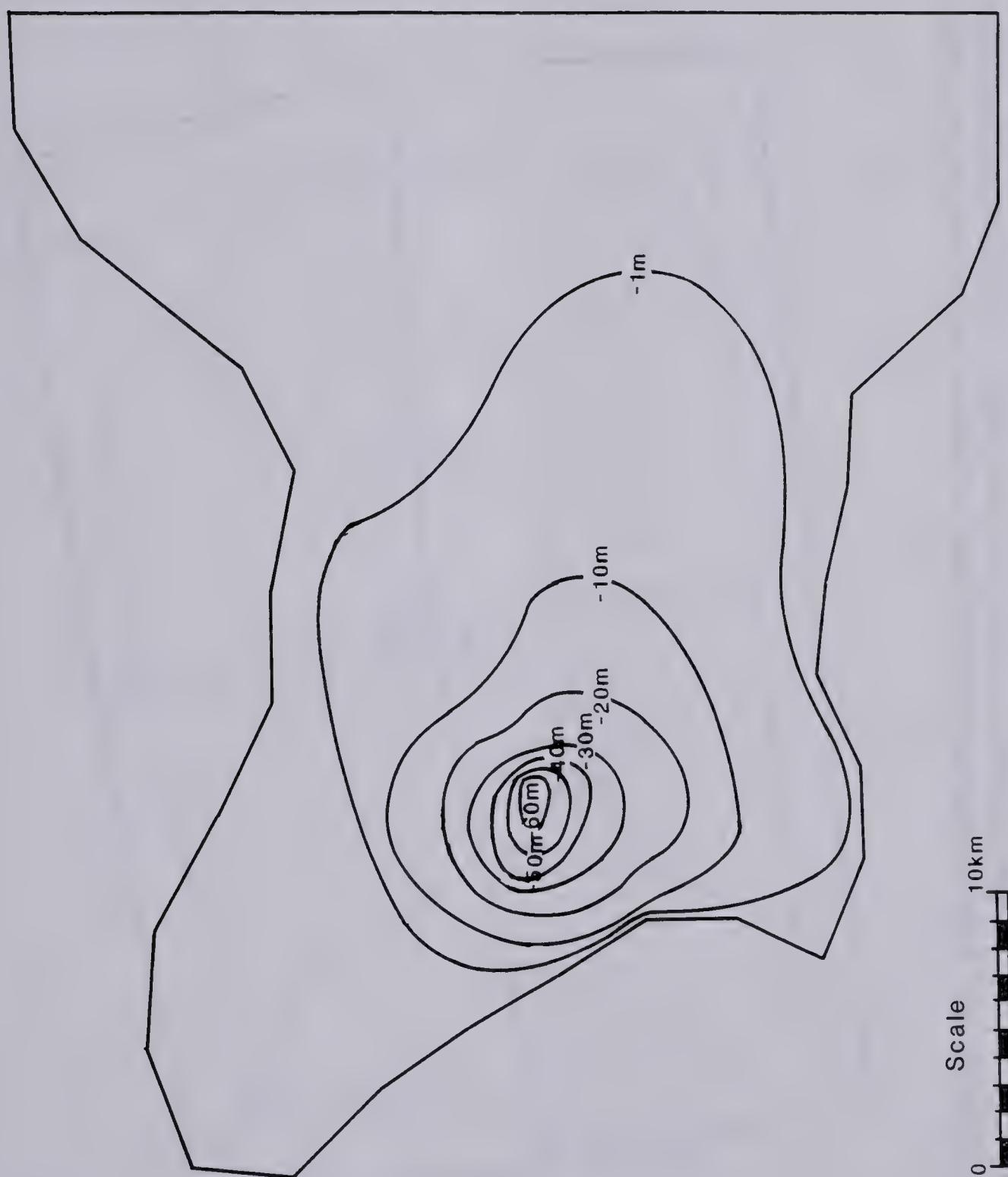


Figure 9. Drawdown cone after 15 years simulation period.

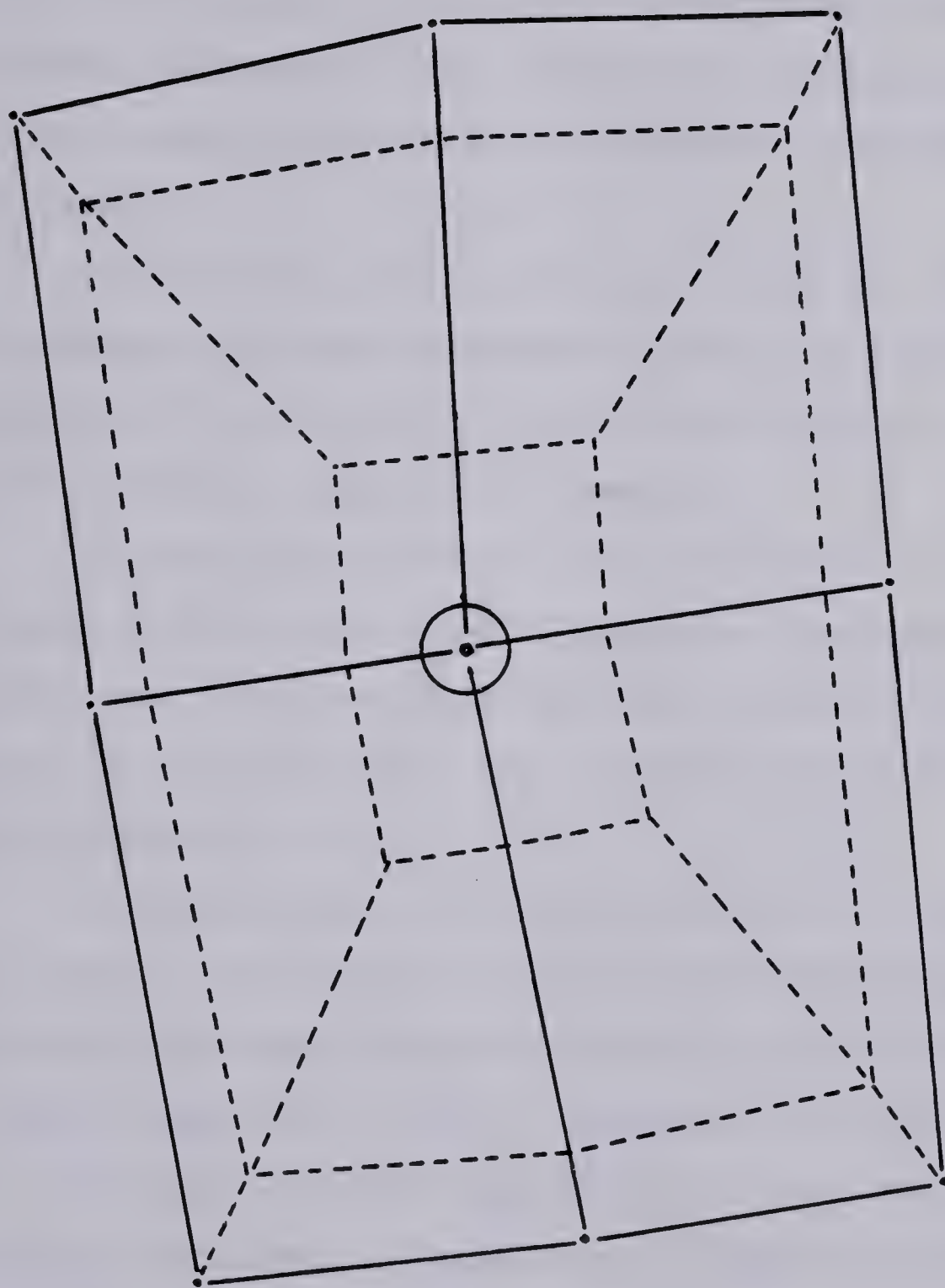


Figure 10. Figure demonstrating how the elements around a flux node are partitioned to give the impression of mesh compression. The solid lines represent the existing elements. The dashed lines represent the partitioning that is later smoothed.

represented by a series of constant head nodes at the surface. The south, bottom and top sides of the system are defined as no-flow boundaries. The simulation of dewatering the open pit mine was accomplished by defining constant head nodes that represent drawdown to the base of the mine. It is assumed that there is no recharge to the system and that the initial head distribution is consistent everywhere within the domain.

To carry out this simulation, Verge's (1975) algorithm for modelling three-dimensional transient groundwater flow was used. The simulation was carried out over a period of 15 years using a time-step of 5 years.

The mesh generated for the simulation is illustrated in Figure 8. The layer shown illustrates the general configuration of the first of eight layers of the mesh; each layer having 672 nodes, 580 elements and numbered to have a half-bandwidth of 89.

The mesh density was determined more on the basis of the memory requirements of the Verge (1975) algorithm when used on the Amdahl 5860 mainframe than on the basis of computational efficiency or detailing the solution.

The mesh required approximately 1.5 hours at the graphics terminal to input the information to construct the mesh. Approximately 32 seconds of CPU time, on the Amdahl 5860 mainframe, is necessary to construct the grid. Approximately half of this computer time was involved with the renumbering of the mesh to reduce the bandwidth. In

addition, approximately another hour of time is required to determine the location of constant head nodes that define the position of the river and the mine.

The drawdown cone, as seen at the surface after 15 years, is illustrated in Figure 9. This drawdown cone follows the outline of the mined area and provides a reasonable representation of what one would expect. The simulation required approximately 45 seconds of CPU time on the Amdahl 5860 mainframe.

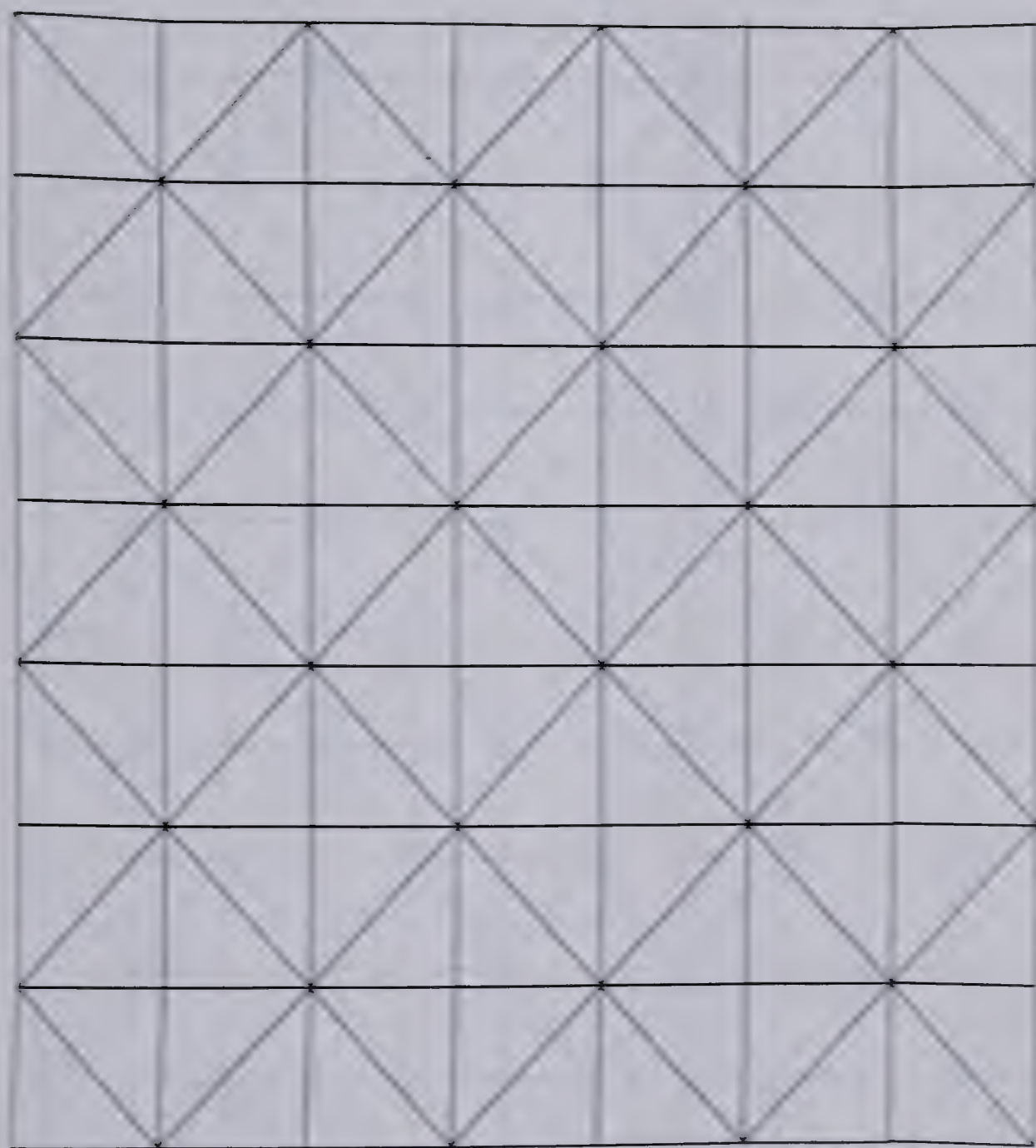


Figure 11a. Basic rectangular shaped figure showing the domain discretized into 196 triangular elements.

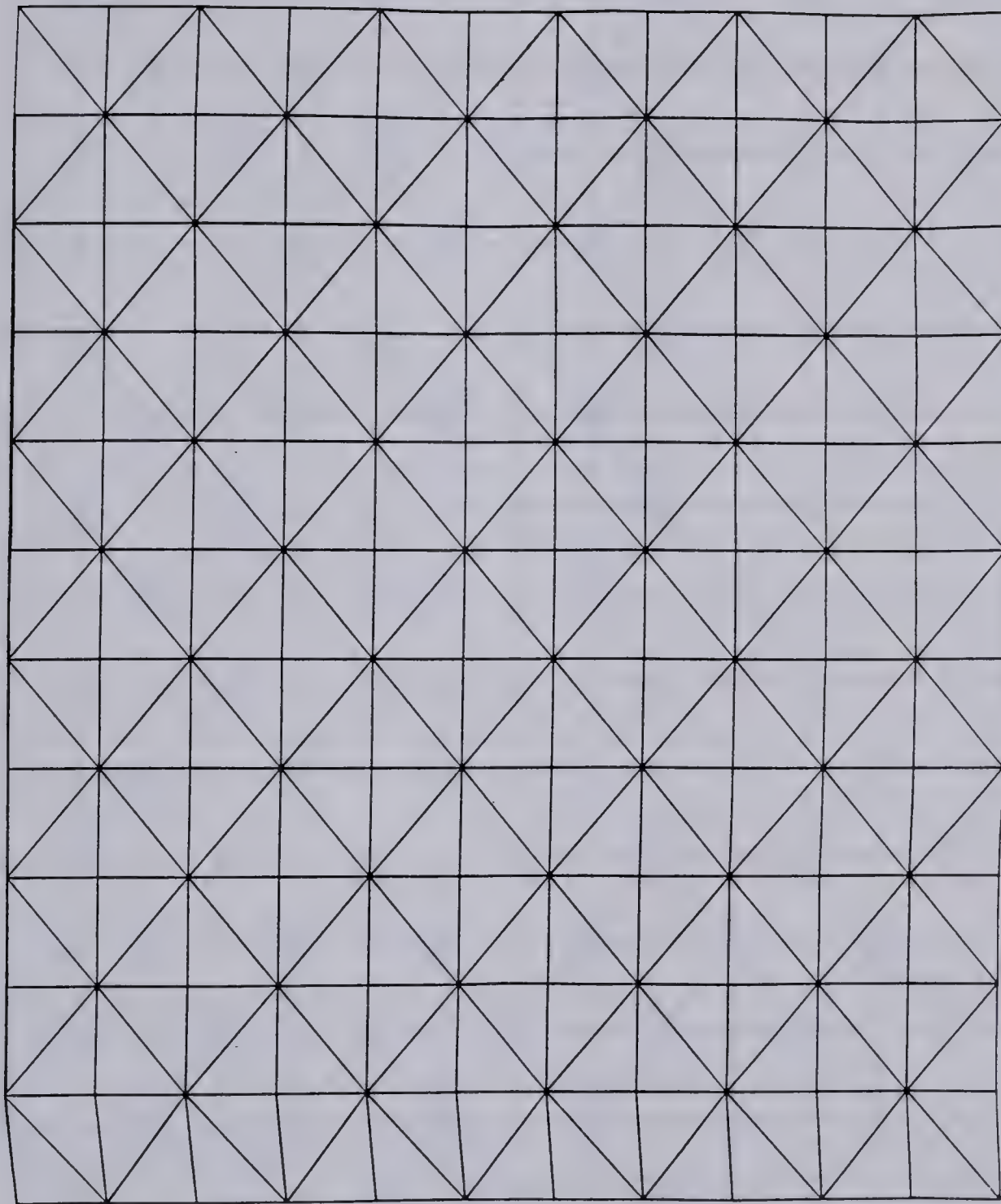


Figure 11b. Basic rectangular shaped figure showing the domain discretized into 484 triangular elements.

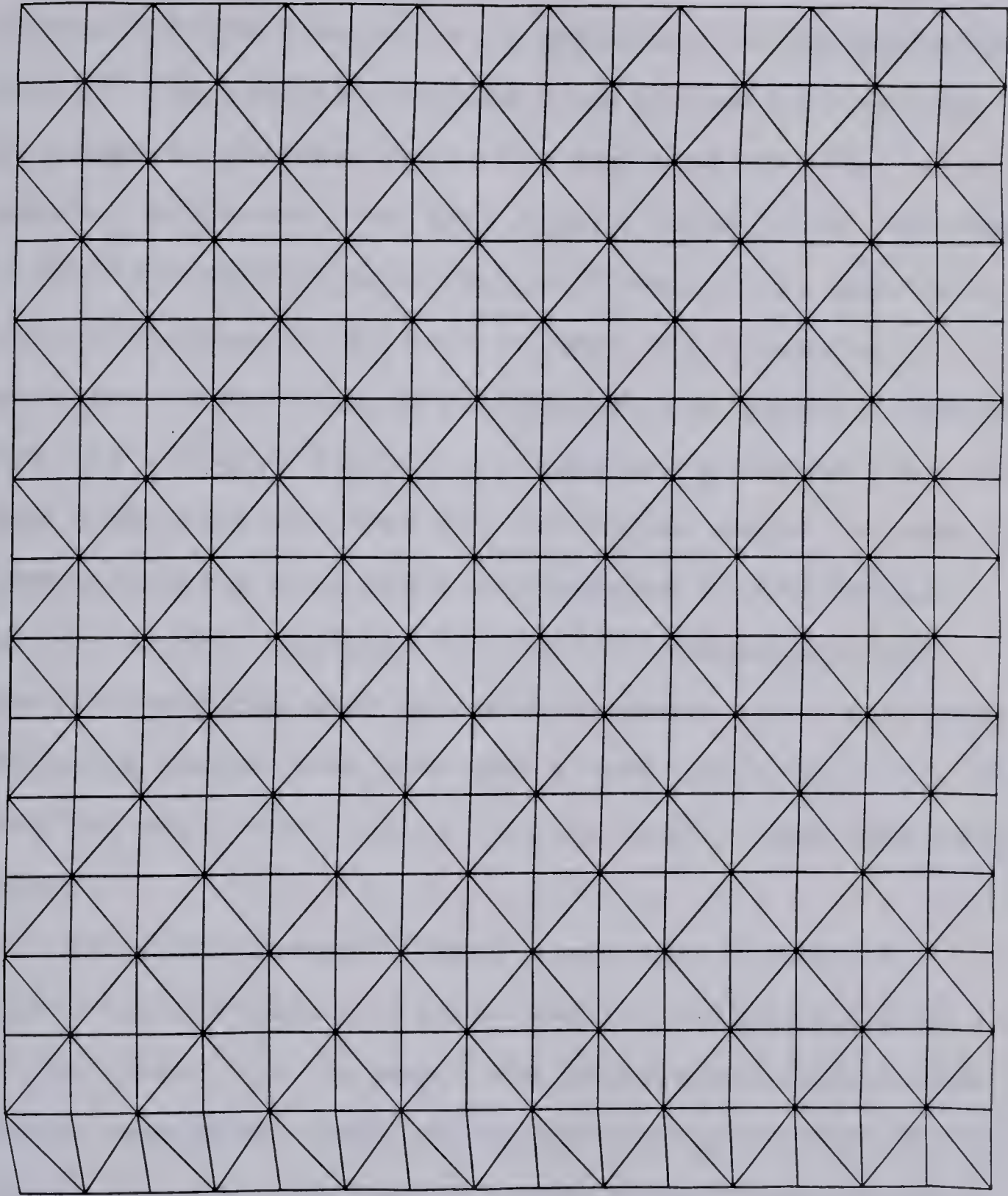


Figure 11c. Basic rectangular shaped figure showing the domain discretized into 900 triangular elements.

4. OPTIMIZATION OF THE MESH GENERATION PROCESS

With the availability of the mesh generation procedure, the problem still remains as to whether the mesh produced is optimum for the simulation in question. In the previous chapter, the results obtained from the hand-drawn mesh and the computer-produced mesh, for the same problem, were somewhat different from each other. Where it is not possible to verify numerical solutions with analytical solutions for a given problem, it is hard to know which simulation is more accurate. In the example considered, the computer generated mesh has a higher density of nodes and elements than the hand drawn version. However, the former could be considered computationally more efficient because of the smaller half-bandwidth. Arguably it could be stated that the computer produced mesh has more elements and nodes, and therefore should have provided a more exact solution. The question would then become "did we really need that many nodes?".

Bathe and Sussman (1983) state that a mesh of relatively few elements whose nodal points are placed in optimal positions can match the predictive capabilities of a coarse mesh whose nodal points are not particularly well placed. As well, the use of higher-order isoparametric elements can improve the solution if the elements were distorted in such a manner that they fit the optimal positions of the nodes. This last point will not be considered further because the generation scheme presented

herein only generates linear elements.

In my code two facilities were incorporated to help optimize the mesh. There is an inherent advantage in dealing with pre-positioned flux nodes in dealing with hydrogeologic simulations. In other words, because the source of the flux is known, it is possible to compress the mesh to maximize mathematical accuracy. The manner in which this compression is performed is illustrated in Figure 10. The existing elements are represented by solid lines, while the partitioning is represented by dashed lines. It can be seen that the elements surrounding a given well (or flux node) are partitioned to form more elements. The partitioned elements are then "smoothed" using an averaging technique to reposition the nodes so that there is a smooth transition in element size away from the flux node. Time constraints did not allow me to investigate grid compression around a flux node more thoroughly. This would have involved optimizing my scheme and investigating other schemes such as that used by Bathe and Sussman (1983).

As well there is a routine to optimize the computational effort involved with the solution of the finite element mesh. This routine renumbers the nodes in order to reduce bandwidth, but unlike the algorithm provided by Collins (1973), the routine incorporated into this procedure renumbers the nodes beginning at one corner of the generated mesh. Collins' (1973) algorithm provides the ability to find the optimum node numbering scheme by

successive renumberings, starting with a different node each time, and comparing the resultant bandwidths. From the solutions provided by Collins (1973), the best schemes begin numbering at a corner of the domain. It is not known whether this provides the optimal renumbering scheme in every case, but it was felt that the extra effort and cost to find the optimal condition would not result in savings in computational effort within the solution.

Logically, the efficiency of the renumbering scheme may be due in part to the geometry of the mesh (ie; there is always a long and a short axis). It is assumed that the worst case situation would be one where the mesh forms a large cube and the bandwidth would be related to the length of a diagonal through the cube. For the vast majority of cases, the renumbered mesh should have a smaller bandwidth than the original.

Workers in the field of structural engineering have come up with schemes to provide optimum meshes for finite element simulations (Bathe and Sussman, 1983; Shephard et al., 1979, 1980). These procedures involve an iterative approach beginning with a crude grid that is optimized when interacted with a finite element solution. In structural engineering this optimization is accomplished through modifying the initial mesh until the strain energy of the solution is maximum or the potential energy is minimized (Bathe and Sussman, 1983). In hydrogeology, this approach would be analogous to modifying the mesh until the change in

the potential head is minimized for each element. A fundamental approach to this problem is that the potential should be minimized not only with respect to finite element nodal displacements but also with respect to the nodal point locations (Bathe and Sussman, 1983). Defining the potential energy by Π , and the condition that the energy not fluctuate by $\partial\Pi=0$, gives the following two equations.

$$\begin{aligned}\partial\Pi/\partial u_i &= 0 \\ \partial\Pi/\partial\beta_i &= 0\end{aligned}\tag{3}$$

where: u_i = the nodal point displacements

β_i = all the nodal coordinates that can be changed without changing the geometry of the domain.

The effect of satisfying these equations is that the resulting mesh is concentrated around areas of greatest flux. In my code there exists a facility to concentrate flux nodes prior to the initial simulation. Therefore, satisfying these equations would involve an iterative approach once the initial solution is performed.

It is beyond the scope of this work to provide algorithms for solutions to these two equations. Indeed it is only recently that workers in this area have begun to

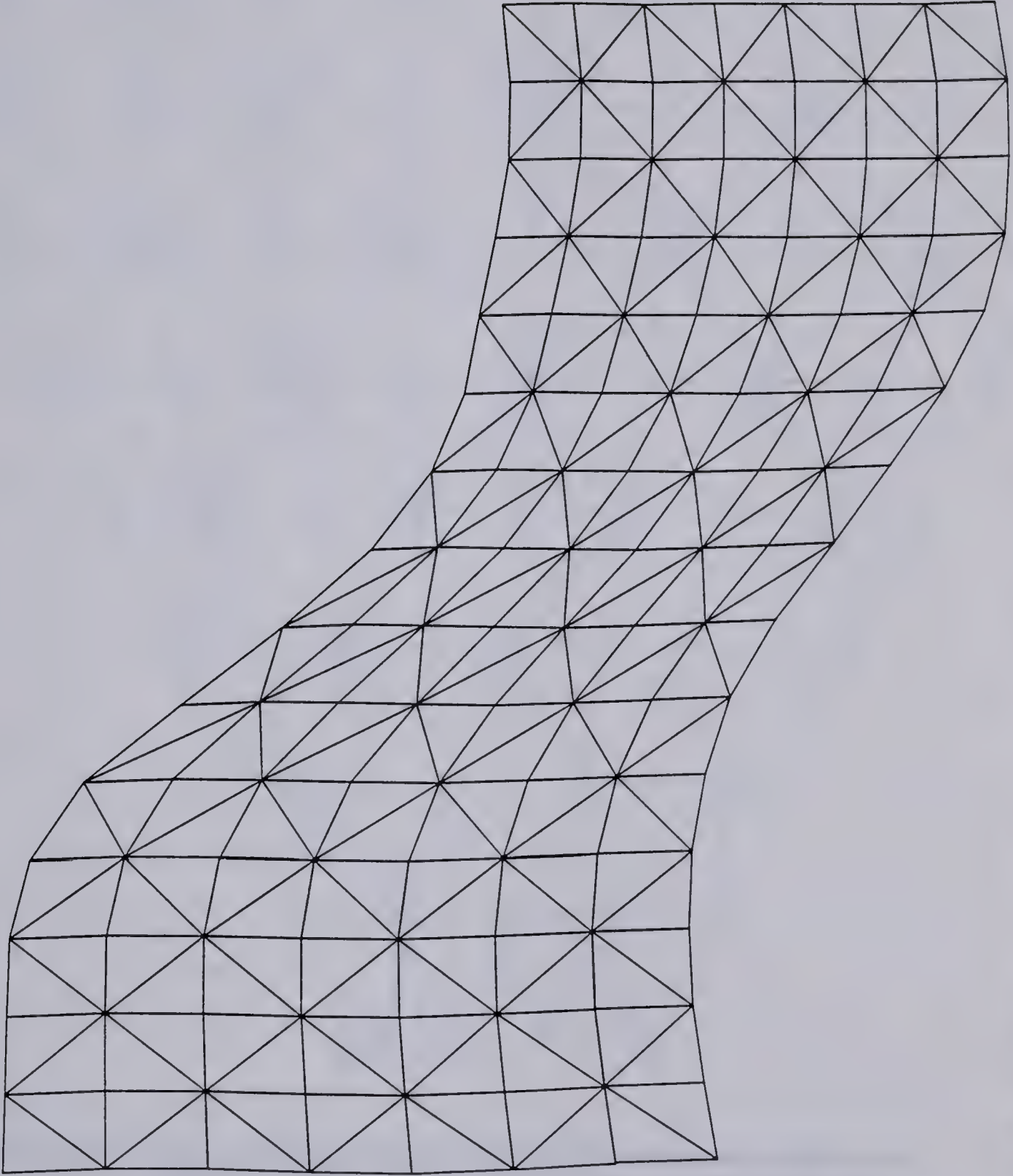


Figure 12. Irregular shaped figure of a type to represent a buried channel.

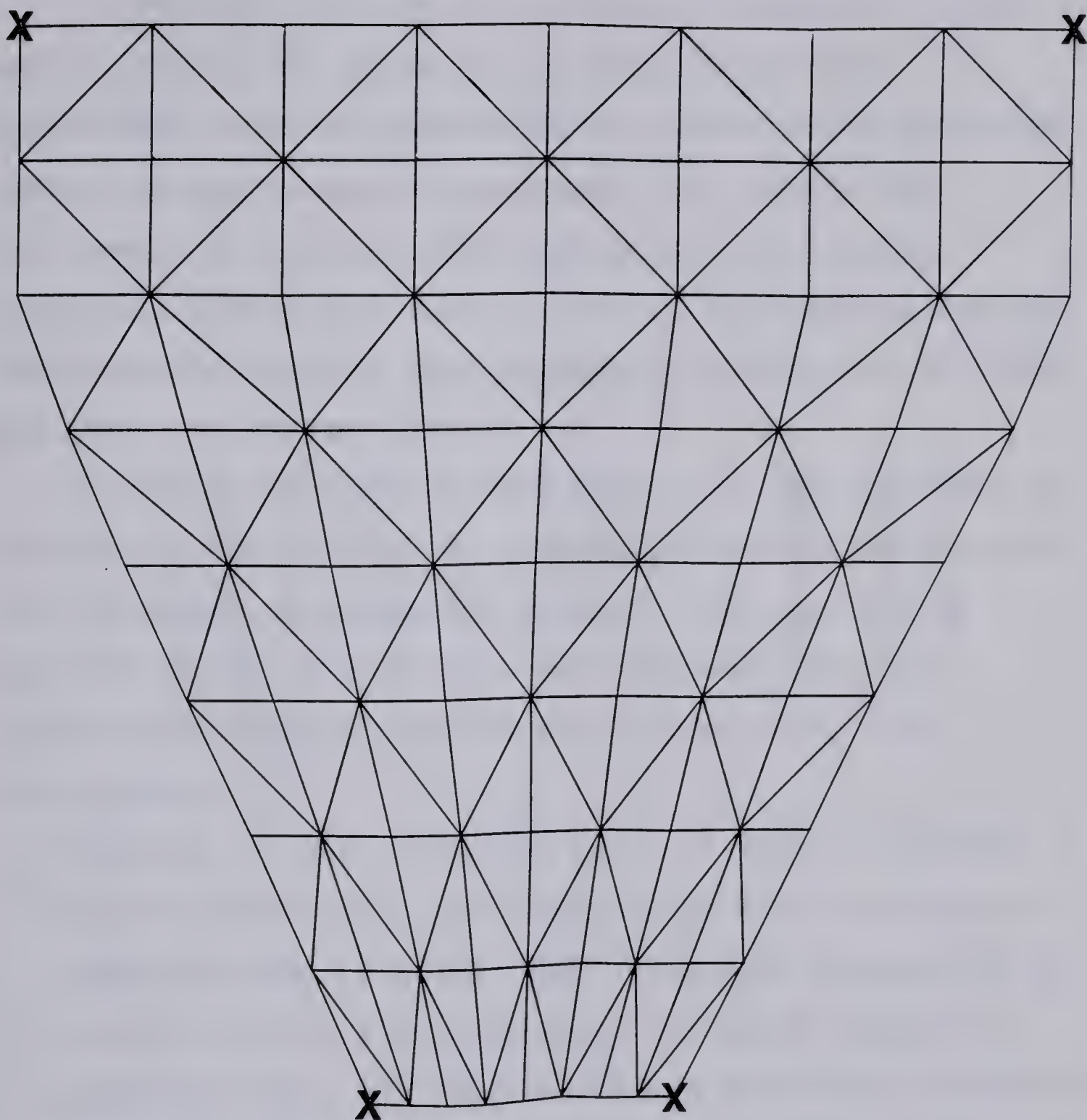


Figure 13. Triangular shaped figure discretized with a mesh of triangular elements. Corner points used to generate the mesh are designated with an "X".

deal with this problem (Bathe and Sussman, 1983; Shephard et al., 1979, 1980).

In my work there has been a modest attempt to provide a degree of built-in optimization. There exist some approximate rules and guidelines for constructing effective meshes through iterative procedures that involve the performance of analysis. The mesh generation scheme presented in this work does not easily allow one to use the approaches outlined by such authors as Shephard et al (1980) and Bathe and Sussman (1983).

It would therefore be advantageous in the synthesis of the mesh to try to bring in a degree of optimization before the simulation is attempted. Arguably this can only be attained through a great deal of experience, but two guidelines are presented with the figures to aid the uninitiated:

1. Probably the most important point is that in problems of aquifer hydraulics, the region of greatest variation in hydraulic head is known. Even though the program has a built-in facility to handle the increased change in hydraulic head, the user is advised that there should be more elements in an area where greatest changes in potential occur, and less where little change is anticipated and they are not necessary.
2. Although previously stated in the body of this work, that in dealing with isoparametric elements, it is not easy to change the density of the elements, there are a

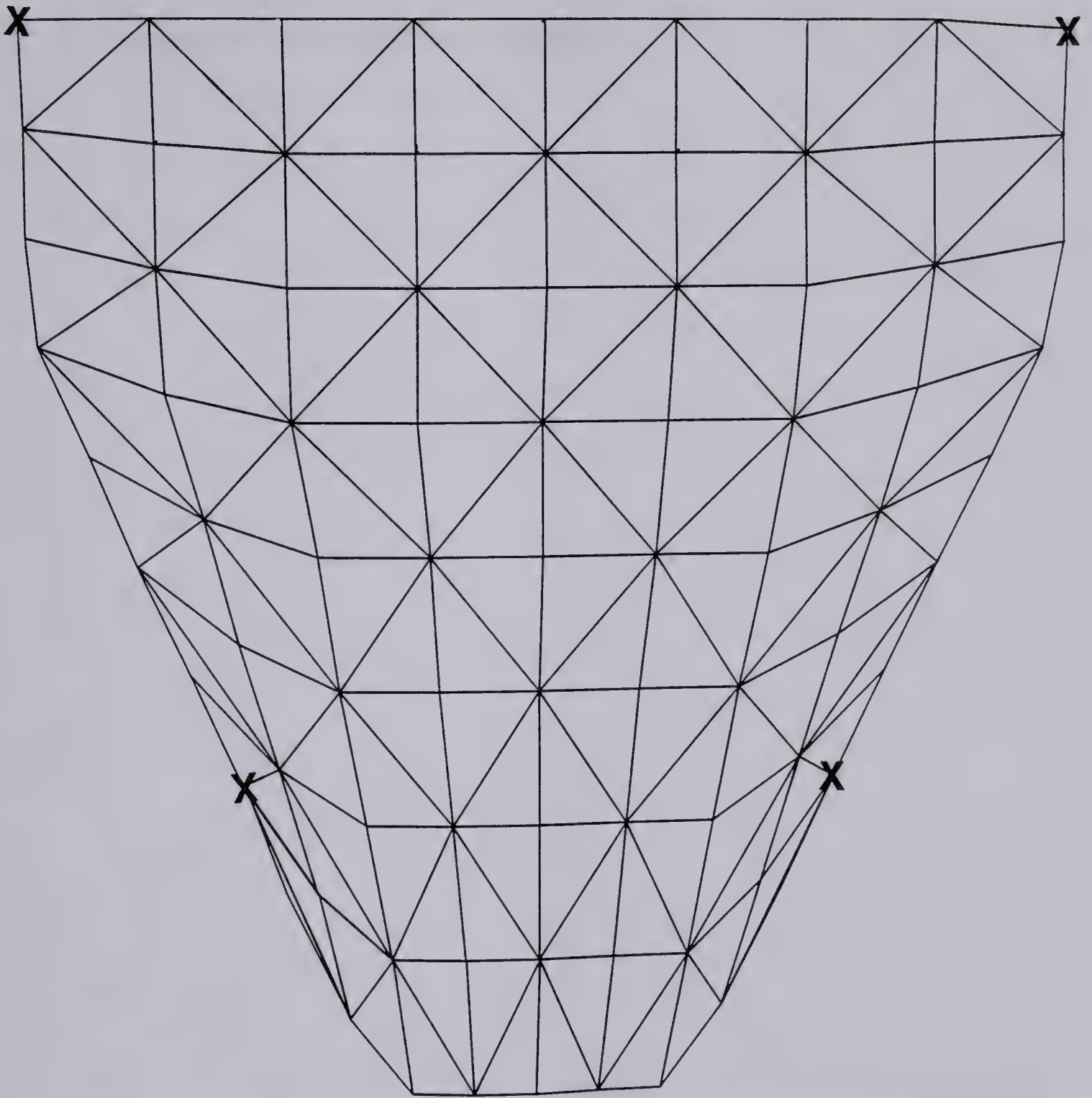


Figure 14. Triangular shaped figure showing results of designating corner points in a different manner. Corner points used to generate this mesh are designated with an "X".

few techniques involved in the placement of corner nodes that can be used to change this distribution. This is illustrated in Figures 13 and 14.

5. CONCLUSIONS AND RECOMMENDATIONS

In accordance with the objectives set out in this study, a finite element mesh generator has been successfully developed. The mesh generation scheme has proven successful and expedient in modelling finite element groundwater flow problems presented herein. It is anticipated that the generator will be successful in generating meshes where the geometry of the domain is not extremely complicated, requiring only gradual changes in mesh density. It will accept large contrasts in hydraulic conductivity, but due to its inability to produce a graded mesh, the solution may be inaccurate. As well, it is anticipated that the scheme will be successful in most cases where the geometry of the domain boundaries do not form very acute angles (ie: generally less than 10 degrees).

The amount of effort normally required to input data for a simulation has been reduced considerably through incorporating into the design of the system a digitizer and interactive graphics terminal. The digitizing tablet is used to specify both numerical information as well as options in mesh generation. The colour graphics terminal, used in conjunction with the Integrated Graphics facilities (Davis, 1977) available at the University of Alberta, allows the user to interact in the mesh generation procedure. The program is capable of generating finite element meshes in both two- and three-dimensions and has proven to be capable of handling problems of the complexity commonly associated

with aquifer hydraulics.

This system operates in a mode that allows the user to be able to generate meshes without having previously absorbed volumous user manuals. The program prompts the user with options throughout the entire generation process. This "user friendly" feature should allow for the wide use of this program.

The graphics features display meshes in a series of eight colours, allowing the user to visualize where he is in the generation process. As well, hard copies can be obtained at any time during the mesh generation process.

The program, by being uniquely developed for the purposes of modelling aquifer hydraulics, has the ability to incorporate both pumping and observation wells within the mesh.

A degree of grid optimization has been incorporated into the program. This capability relates to two main areas, namely: in improving mathematical accuracy; and computational efficiency. To aid in the area of mathematical accuracy, the code compresses the mesh around flux nodes to better approximate rapid changes in hydraulic head around pumping wells. Computational efficiency is aided through the incorporation of the algorithm provided by Collins (1973) to decrease bandwidth.

Recommended areas for further study are listed below.

1. The area of grid optimization should be fully investigated. Again, as in the area of finite element

mesh generation, the field of structural engineering provides the logical starting point due to the material already existing on this problem in that field. It is suggested that the work involve using the information already supplied as input to the program such as transmissivity, storativity and pumping rates and the relationships of potential energy as provided by Bathe and Sussman(1983) as a logical starting point.

2. With the advent of the personal computer and their increasing memory capabilities and decreasing costs, it would be advantageous to investigate the feasibility of adapting this system for use with these smaller systems so as to readily allow its use by small engineering companies to aid in the speedy solution of everyday finite element simulation problems. Problems envisioned in this would be adapting the program's graphics to the capabilities of the personal computer's graphics and decreasing storage requirements for this program.

REFERENCES

- Bathe, R.J. and Sussman, T.D., (1983), An Algorithm for the Construction of Optimal Finite Element Meshes in Linear Elasticity. In Computer Methods for Non-Linear Solids and Structural Mechanics. A.S.M.E. Applied Mechanics Division, vol 54, Ed. Satya N. Atluri.
- Cavendish, J.C., (1974), Automatic Triangulation of Arbitrary Planar Domains for the Finite Element Method. Int'l J. Numerical Methods Eng. vol 8, pp 679-696.
- Collins, R.J., (1973), Bandwidth Reduction by Automatic Renumbering. Int'l J. Numerical Methods Eng. vol 6, pp 345 356.
- Davis, A.R., (1977), Integrated Graphics User's Guide. Computing Services, University of Alberta, Edmonton, Alberta.
- Hornbeck, R. W., (1975), Numerical Methods. Quantum Publishers, Inc. New York, N.Y. 310 pages.
- Klienstreurer, C., (1980), Interactive Finite Element Mesh Generators for Flow Systems. Finite Elements in Water Resources, vol 3, pp7.99 - 7.109.
- Narasimhan, T.N. and Witherspoon, P.A., (1980), Overview of the Finite Element Method in Groundwater Hydrology. Finite Elements in Water Resources, vol 3, pp1.29 - 1.44.
- Sadeck, E.A., (1980), A Scheme for the Automatic Generation of Triangular Finite Elements. Int'l J. Numerical Methods Eng., vol 15, pp 1813-1822.

- Sartori, L., Aurelio, V., Riccioni, M., and Svarca, G., (1982), Automatic Generation of Finite Element Network for Bi-Dimensional and Quasi-Three-Dimensional Equations Governing the Groundwater Flow. Finite Elements in Water Resources., vol 5, pp 8.13 - 8.23.
- Shephard, M.S., Gallagher, R.H. and Abel, J.F., (1979), Experience with Interactive Computer Graphics for the Synthesis of Optimal Finite Element Meshes. In Finite Element Grid Optimization, The Third National Congress on Pressure Vessels and Piping, A.S.M.E. PVP-38, 73 pages.
- Shephard, M.S., Gallagher, R.H. and Abel, J.F., (1980), The Synthesis of Near-Optimum Finite Element Meshes with Interactive Computer Graphics. Int'l J. Numerical Methods Eng., vol 15, pp 1021-1039.
- Suhara, J. and Fukuda, J., (1972), Automatic Mesh Generation for Finite Element Analyses. Advances in Computational Methods in Structural Mechanics and Design, (Ed. Oden, J.T., Clough, R.W. and Yamamoto, Y.) U.A.H. Press, Huntsville, Alabama.
- Thwacker, W.C., (1980), A Brief Review of Techniques for Generating Irregular Computational Grids. Int'l J. Numerical Methods Eng., vol 15, No. 9, pp 1335-1341.
- Verge, M.J., (1972), Linear Triangular Finite Element Model for Horizontal Aquifer Analysis with Nodal Transmissivity Values. Unpublished Paper, University of Waterloo, Department of Earth Sciences, Waterloo,

Ontario, Canada.

Verge, M.J., (1975), A Three-Dimensional

Saturated-Unsaturated Groundwater Flow Model for
Practical Applications. Unpublished Ph.D. Thesis,
Department of Earth Sciences, University of Waterloo,
Waterloo, Ontario, Canada.

Yerry, M.A. and Shephard, M.S., (1983), A Modified Quadtree
Approach to Finite Element Mesh Generation. Computer
Graphics and Applications., pp 39 46.

Zienkiewicz, O.C. and Phillips, D.V., (1971), An Automatic
Mesh Generation Scheme for Plane and Curves Surfaces by
Isoparametric Coordinates. Int'l J. Numerical Methods
Eng., vol 3, no. 4, pp 519-528.

APPENDIX A User's Guide

To use the automatic mesh generation program one should have at his disposal: a diagram of the figure to be modelled, with transmissivity zones, pumping wells, observation wells and areas of leakance and values of necessary constant head delineated; and hydrogeological parameters. It is also assumed that the user has the computer hardware or equivalent previously described. To run the compiled version in MTS the commands are: \$RUN

MESH.OBJ+*IG 5=*MSOURCE* 6=*SINK* 7=-A 9=-C

where: MESH.OBJ is the compiled version of the automatic mesh generation program.

*IG is the integrated graphics library of subroutines available at the University of Alberta.

5=*MSOURCE* specifies the input port (or device) that accepts data from the keyboard and digitizing tablet.

6=*SINK* specifies the terminal as the output port. This facilitates prompting, querying and displaying the results so that the user is involved with all aspects of the mesh generation. *SINK* displays everything on the terminal screen.

7=-A this is the port(or device) where the resultant mesh information is outputted to after the generation is completed.

9=-C this is the port to which all plots requested are sent for plotting on the CalComp Plotter.

Once the program is loaded the user will be asked to respond to a series of queries. This section is intended to aid the user as to what to expect to be able to answer as the generation routine progresses.

1. IM-a variable that determines whether one is using English or metric units. IM=0 informs the program that the user is working in English units and the program is expecting to digitize a map where "1 inch=x feet". If IM=1, the program will convert all coordinates into meters. The program is expecting to digitize a map where the scale is defined as "1:xxxx".
2. Scale-when entering the scale the format is F10.0. Therefore the user is cautioned to remember to input a period(or a comma when dealing with integer input) after the number is typed in such as 10. or 10,.
3. ID-a variable that specifies whether one is working with a two-dimensional or three-dimensional mesh. ID=0 implies that the mesh to be generated is two-dimensional and to be generated in the x-y plane. ID=1 implies the mesh to be generated is three-dimensional, generated in the x-y plane and projected in the z-direction.
4. IELEM-prompts the user as to what type of element is required. In two-dimensions, the request would be for IELEM=0, which implies triangular elements, or IELEM=1 for quadrilateral elements. In three-dimensions the choice would be between IELEM=0 for tetrahedral elements, or IELEM=1 for parallelepiped (or "brick-type")

elements. Note that this mesh generator is only capable of constructing linear elements.

5. ISS-is a prompt only used when one is working in 3-D. Where ISS=0 implies Steady State and ISS=1 implies Transient simulation. This outputs a value of 999. for the time, that triggers the steady state simulation in Verge's (1975) 3-D Groundwater model program.
6. This prompt requests the user to indicate the "LOWER LEFT" "LOWER RIGHT" and "UPPER RIGHT" corners of the map on which the flow domain is defined. These data are provided so that the computer can scale the drawing for purposes as displaying on the screen, plotting and for determining whether the map is tilted with respect to the axis of the digitizer.
7. The next prompt is the menu. The user is provided with 11 options that would control the program in the manner outlined in the flow chart in Figure 4. The following prompts will be in the order as followed by this flow chart, and a progression through the mesh synthesis process.
8. After selecting Switch No. 9, the user would be asked to enter the storativity and the transmissivity (in 3-D the hydraulic conductivities). The format for reading the values is E10.3. After this, the user would be requested to digitize the perimeter of the transmissivity zone in a counter-clockwise manner.
9. If any pumping wells are present, the user would then

select Switch No. 7. In 2-D the prompt would be to enter the "start time", "finish time", and the number of time steps for the simulation, and the pumping rate in L^3/sec . The query would then be asked whether there is any more pumping information. This question provides the opportunity to add more simulation periods as is compatible with the two-dimensional Verge-Frind (Verge, 1972) transient code for confined aquifers. In three-dimensions the user would be requested to enter the simulation period, (the time units are dependent on the time units designated in the Verge(1975) three-dimensional model), initial time step, a multiplier for the time step, number of time steps changes in delta time and the pumping rate in L^3/sec . The format for entering the start time, finish time, simulation period, initial time step, multiplier for the time step and pumping rate are all F10.0. The formats for entering the number of time steps and the number of time steps between changes in delta time are I4.

10. The next prompt would be encountered after selecting Switch No. 13 for generating a mesh. The prompt would be to designate the four (eight in three-dimensions) corner points of the figure displayed on the terminal. If one is working in three-dimensions, a prompt would request the elevation of the corner point. As well there are prompts that request that the user double check the points that he has input. The next prompt would have the

user specify the number of elements desired in the x-, y- and z-directions to obtain the desired mesh density.

11. Once the mesh is drawn, the user is prompted as to whether or not the figure displayed on the screen is to be plotted on the CalComp Plotter. From here, the user has the option of changing the size of the plot, with the default being the same size as that displayed on the screen.

12. Other Prompts include:

Entering the leakance values in format E10.4.

Entering the boundary fluxes in format F10.0.

Dimensioning of Program Arrays

As noted in the body of the text, the arrays are all dimensioned sufficiently large for most situations. However the following is a guide should there arise a need to change the storage requirements.

```
COMMON MEMJT(NNODE,32),IN(NELM,8),IFACE(3500,6),YN(NNODE),
      ZN(NNODE),CX(NNODE),XN(NNODE),JMEM(NNODE),NODE1(8),
      NDX(NNODE),NDY(NNODE),NDZ(NNODE),ITYPE(NNODE),
      NDWEL(20,2,25),LEAK(100),ALEAK(100),XIN(100),
      YIN(100),ISAVE(25,2),PPOSX(10),PPOSY(10),PPOSZ(10),
      OBSX(10),OBSY(10),OBSZ(10),ND(16),NND(16),COX(10),
      COY(10),COZ(10),IFE(6),NPMP(25),IPPN(10),ICRNR(10,8),
      NFACE(6,4),ICNTST(4,2),NBDND(4000)

DIMENSION TIME(10,2),NTIME(10),PMP RTE(10,10),X1(4),Y1(4),
```


Z1(4),TITLE(10),BFLUX(10),A(3),B(3),NCEL(10,2),
CHNG(10),ITCHNG(10),STOR(10)

INTEGER COL(10)

Where: NNODE=estimated number of nodes.

NELM =estimated number of elements.

Use of arrays in common:

MEMJT - stores the nodes connected to each node. Used
in SETUP and OPTNUM.

IN - array describing incidence of each element.

IFACE - array denoting elements appearing in each of
the six faces of the mesh. Used throughout
the program.

XN - array storing x-coordinate of each node.

YN - array storing y-coordinate of each node.

ZN - array storing z-coordinate of each node.

CX - transmissivity of each node.

JMEM - number of nodes connected to each node. Used
in SETUP and OPTNUM.

NBDND - nodes that form the perimeter of each
transmissivity zone.

NDX - the adjacent node in the positive x-direction.

NDY - the adjacent node in the positive y-direction.

NDZ - the adjacent node in the positive z-direction.

ITYPE - array denoting type of each node,

ie: 0-normal

1-constant head

2-flux node.

NDWEL - defines the node numbers for observation and pumping wells in 2-D and 3-D. Dimensioned for 20 wells, observation or pumping well, and for each well to occupy 25 nodes in a vertical sense.

LEAK - defines an element that a leakance value is assigned to.

ALEAK - defines the leakance value for each element.

XIN - for inputting x-coordinates in "Pic" routines.

YIN - for inputting y-coordinates in "Pic" routines.

ISAVE - defining nodes that surround a pumping well.

PPOSX - x-coordinate of a pumping well.

PPOSY - y-coordinate of a pumping well.

PPOSZ - z-coordinate of a pumping well.

OBSX - x-coordinate of an observation well.

OBSY - y-coordinate of an observation well.

OBSZ - z-coordinate of an observation well.

ND - used in the concentration of elements around a pumping node.

NND - used in the concentration of elements around a pumping node.

COX - hydraulic conductivity in the x-direction.

COY - hydraulic conductivity in the y-direction.

COZ - hydraulic conductivity in the z-direction.

IFE - used as a counter in conjunction with IFACE.

NPMP - stores the number of nodes associated with each well.

IPPN - stores the number of simulation periods associated with each well.

ICRNR - stores the corner points associated with each transmissivity zone.

NFACE - node numbers with each face of each new transmissivity zone.

ICNTST- store common corner points.

NODE1 - book keeping array for the formation of the elements.

APPENDIX B: Program Listing


```

1 C*****
2 C*****
3 C***
4 C*** GREGORY SMITH M.Sc. THESIS PROJECT
5 C***
6 C*****
7 C*****
8 C
9 C A 2&3D Automatic finite element mesh generator.
10 C This program digitizes information from maps and
11 C cross-sections using a Summagraphics(Corp) ID
12 C series tablet digitizer.
13 C The system is user interactive, with the computer
14 C giving prompts and options to the user as to
15 C what to do next.
16 C Options: 2 or 3D mesh
17 C   triangular(tetrahedral) or Quadrilateral
18 C   (parallelepiped) elements
19 C   metric or English measurement system
20 C The elements are generated through an isoparametric
21 C method(Zienkiewicz and Phillips, 1971) with the
22 C density of the elements determined by the user.
23 C The elements are concentrated around pumping
24 C wells for numerical stability.
25 C After the mesh is generated and displayed and all
26 C manipulations are complete, the nodes are then
27 C renumbered to reduce bandwidth(Collins, 1973),
28 C then output for use by a F.E. program.
29 C In 3D the mesh should be generated from the
30 C bottom up.
31 C
32 C .....
33 C
34 C Program MESH.GEN
35 C
36 C Author
37 C   G.J. Smith(U. of Alberta)
38 C
39 C Last Update
40 C   May 1984
41 C
42 C Purpose
43 C   To set up conditions for mesh generation
44 C   call up the various routines to
45 C   generate the mesh and display and output
46 C   the resultant mesh.
47 C
48 C Usage
49 C   Execute as a main Program
50 C

```



```

51 C
52 C      For complete description see:
53 C      Smith, G.J., Finite Element Pre-processor
54 C      for Hydrogeologic Modelling. Unpublished
55 C      M.Sc. Thesis, University of Alberta, 1984.
56 C
57 C      Subroutines Required
58 C      SMOOTH,DSTRB,NODE,ELMNT,NDEGEN,INTANG.
59 C      SETUP,OPTNUM,ASKPLT,CNCTQ,CNCTT1,CNCTT2,
60 C      Integrated Graphics Library Subroutines
61 C
62 C      .....
63 C      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
64 C      1 ZN(5000)
65 C      COMMON CX(5000)
66 C      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
67 C      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
68 C      1 ALEAK(100)
69 C      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
70 C      1 PPOSZ(10)
71 C      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
72 C      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
73 C      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
74 C      COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
75 C      1 XC, IPN
76 C      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
77 C      1 DELTZ
78 C      DIMENSION TIME(10,2), NTIME(10), PMPRTE(10,10), X1(4), Y1(4),
79 C      1 Z1(4)
80 C      DIMENSION TITLE(20), BFLUX(5000), IBF(1000), A(3), B(3)
81 C      DIMENSION NCEL(10,2), CHNG(10), ITCHNG(10), STOR(10)
82 C      INTEGER YES /'Y'/, NO /'N'/
83 C      INTEGER COL(10) /'RED','BLUE','GREEN','MAGE','CYAN','WHIT',
84 C      1 'YELL','RED','BLUE','GREEN'/
85 C      INTEGER INN(8) /5, 6, 7, 8, 1, 2, 3, 4/
86 C      DATA BETA /0.00E+00/, DENS /0.998/, HETA /0.35/, TETA /0.35/, AL
87 C      1 /0.0/
88 C      Initialize all counters
89 C      10 NLN = 0
90 C      NBN = 0
91 C      IW1 = 0
92 C      IW2 = 0
93 C      IP = 0
94 C      NPP = 0
95 C      IPL = 0
96 C      NPTS1 = 1
97 C      NPTS1 = 1
98 C      ICON = 0
99 C      IPP = 0
100 C      IPN = 0

```



```

101 ITRNS = 0
102 NBF = 0
103 ITRNSS = 0
104 IO = 0
105 NELM = 0
106 NELM1 = 1
107 IOO = 0
108 DO 20 I = 1, 6
109   IFE(I) = 0
110 20 CONTINUE
111 DO 30 I = 1, 4999
112   NDX(I) = 0
113   NDY(I) = 0
114   NDZ(I) = 0
115   ITYPE(I) = 0
116   BFLUX(I) = 0.
117 30 CONTINUE
118 DO 60 I = 1, 25
119   DO 50 J = 1, 20
120     DO 40 K = 1, 2
121       NDWEL(J,K,I) = 0
122     40 CONTINUE
123   50 CONTINUE
124 60 CONTINUE
125 C Begin interactive input of options
126   WRITE (6,70)
127 70 FORMAT ('Enter a title for this simulation')
128   READ (5,1890) TITLE
129   WRITE (6,80)
130 80 FORMAT ('***** 2-D - 3-D FINITE ELEMENT MESH GENERATOR *****', /,
131 1   'This Program will operate in two modes; 2-D and 3-D.', /,
132 2   'Some Preliminary information is requires first before', /,
133 3   'proceeding, Enter the frame units - IM -,', /,
134 4   'IM=0 frame units are in inches', /,
135 5   'IM=1 frame units are in centimeters', /,
136 6   '//, ' These are the only acceptable frame units', /,
137 7   ', the format is I I1')
138   READ (5,90) IM
139 90 FORMAT (I1)
140   WRITE (6,100)
141 100 FORMAT ('Scale:?' Enter the scale of the figure to b digitized')
142   IF (IM .EQ. 1) GO TO 130
143   WRITE (6,110)
144 110 FORMAT ('1 inch= ?ft. ')
145   READ (5,120) SCL
146 120 FORMAT (F10.0)
147   GO TO 150
148 130 WRITE (6,140)
149 140 FORMAT ('1: ?')
150 150 READ (5,120) SCL

```



```

201 READ (5,290) IA, IB
202 FORMAT (2X, 2I6)
203 A(I) = FLOAT(IA)
204 B(I) = FLOAT(IB)
205
206 CONTINUE
207 WRITE (6,310) IM, A(1), B(1), A(2), B(2), A(3), B(3)
208 FORMAT ('IM= ', I1, /, 'A(1)= ', F6.0, 4X, 'B(1)= ', F6.0, /,
209 'A(2)= ', F6.0, 4X, 'B(2)= ', F6.0, /, 'A(3)= ', F6.0, 4X,
210 'B(3)= ', F6.0)
211 DIFX = A(2) - A(1)
212 DIFY = B(3) - B(1)
213 SLOPEV = DIFY
214 IF (A(1) .NE. A(3)) SLOPEV = DIFY / (A(3) - A(1))
215 SLOPEH = (B(2) - B(1)) / DIFX
216 ANGLE = ARSIN(SLOPEH)
217 TUM = 1.
218 IF (B(1) .EQ. B(2) .AND. A(1) .EQ. A(3)) TUM = 0.
219 SCALE = 1000.
220 IF (IM .EQ. 1) SCALE = 393.7
221 ICK = 1
222 N = 0
223 YC = ((B(3) - B(1))/2.)*SCL) / SCALE
224 XC = ((A(2) - A(1))/2.)*SCL) / SCALE
225 ZC = YC * 2.
226 OX = 0000.
227 OY = 0000.
228 GO TO 350
229
230 WRITE (6,340)
231 FORMAT ('A ZERO OR NEGATIVE VALUE IS INVALID', /, ' TRY AGAIN')
232
233 C Display of program options in menu format
234
235 NOW BEGIN DIGITIZING', //,
236 MENU', /, '-----', //,
237 1 | 13 | Generate Mesh(after digitizing)', /, '-----', /,
238 2 | 12 | Output Data for Later Use', /, '-----', /,
239 3 | 11 | Input boundary fluxes', /, '-----', /,
240 4 | 10 | Future Use', /, '-----', /, 9 | Digitize a T
241 5 | 8 | Digitize an Obse
242 6 | 7 | Digitize a Pumping well',
243 7 | 6 | "Pic" Constant Head Nodes', /,
244 8 | 5 | "Pic" Leakance Elements', /, '-----', /,
245 9 | 4 | Remove Last Transmissivity Zone Generated', /,
246 10 | 3 | Remove Last Point Digitized', /,
247 11 | 2 | Stop/Begin Again', /, '-----', /,
248 12 | 1 | Re-Display Menu', /, '-----', /,
249
250 WRITE (6,370)
251 FORMAT ('| 5 | "Pic" Leakance Elements', /, '-----', /,
252 | 4 | Remove Last Transmissivity Zone Generated', /,
253 | 3 | Remove Last Point Digitized', /,
254 | 2 | Stop/Begin Again', /, '-----', /,
255 | 1 | Re-Display Menu', /, '-----', /,
256
257 READ (5,390,END=2260) IX, IY
258 FORMAT (2X, 2I6)
259 IF (IX .GT. 1000) GO TO 430

```



```

251 ICK = IY / 1000 + 1
252 WRITE (6,400) ICK
253
254 400 FORMAT ('SWITCH NUMBER IS', 2X, I2, '/')
255 GO TO (500, 510, 540, 570, 910, 580, 1030, 1190, 1230, 1370,
256 11380, 1590, 2000), ICK
257 410 WRITE (6,420)
258 420 FORMAT ('SWITCH INVALID, 1<SWITCH<14, TRY AGAIN.')
259 GO TO 380
260
261 C
262 C
263 C TRANSFORM X AND Y CO-ORDINATES TO CORRECT ANGLE
264 430 IF (TUM .EQ. 0.) GO TO 440
265 XX = FLOAT(IX)
266 YY = FLOAT(IY)
267 YTEMP = SLOPEH * (XX - A(1)) + B(1)
268 YY1 = (YY - YTEMP) * COS(ANGLE)
269 Y = ((YY1)/SCALE) * SCL
270 IF (B(2) .EQ. B(1)) Y = ((FLOAT(IY) - B(1))/SCALE) * SCL
271 XTEMP = (YY - B(1)) / SLOPEV + A(1)
272 XX1 = (XX - XTEMP) * COS(ANGLE)
273 X = ((XX1)/SCALE) * SCL
274 GO TO 460
275
276 440 X = ((FLOAT(IX) - A(1))/SCALE) * SCL
277 Y = ((FLOAT(IY) - B(1))/SCALE) * SCL
278 450 FORMAT (2F11.3)
279 460 GO TO (490, 490, 490, 490, 490, 490, 1180, 1220, 1360, 490, 490,
280 1490, 490), ICK
281
282 C
283 C LOADING ARRAY FOR FLINE SUBROUTINES
284 C
285 C
286 C
287 C
288 C
289 C
290 C
291 470 WRITE (6,480)
292 480 FORMAT ('ARRAY IS FULL')
293 490 GO TO 350
294
295 C
296 C SWITCH NO. 1
297 C
298 C
299 C 500 GO TO 350
300 C
301 C
302 C SWITCH NO.2
303 C
304 C
305 C
306 C
307 C
308 C
309 C
310 C
311 C
312 C
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
349 C
350 C
351 C
352 C
353 C
354 C
355 C
356 C
357 C
358 C
359 C
360 C
361 C
362 C
363 C
364 C
365 C
366 C
367 C
368 C
369 C
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
399 C
400 C
401 C
402 C
403 C
404 C
405 C
406 C
407 C
408 C
409 C
410 C
411 C
412 C
413 C
414 C
415 C
416 C
417 C
418 C
419 C
420 C
421 C
422 C
423 C
424 C
425 C
426 C
427 C
428 C
429 C
430 C
431 C
432 C
433 C
434 C
435 C
436 C
437 C
438 C
439 C
440 C
441 C
442 C
443 C
444 C
445 C
446 C
447 C
448 C
449 C
450 C
451 C
452 C
453 C
454 C
455 C
456 C
457 C
458 C
459 C
460 C
461 C
462 C
463 C
464 C
465 C
466 C
467 C
468 C
469 C
470 C
471 C
472 C
473 C
474 C
475 C
476 C
477 C
478 C
479 C
480 C
481 C
482 C
483 C
484 C
485 C
486 C
487 C
488 C
489 C
490 C
491 C
492 C
493 C
494 C
495 C
496 C
497 C
498 C
499 C
500 C
501 C
502 C
503 C
504 C
505 C
506 C
507 C
508 C
509 C
510 C
511 C
512 C
513 C
514 C
515 C
516 C
517 C
518 C
519 C
520 C
521 C
522 C
523 C
524 C
525 C
526 C
527 C
528 C
529 C
530 C
531 C
532 C
533 C
534 C
535 C
536 C
537 C
538 C
539 C
540 C
541 C
542 C
543 C
544 C
545 C
546 C
547 C
548 C
549 C
550 C
551 C
552 C
553 C
554 C
555 C
556 C
557 C
558 C
559 C
560 C
561 C
562 C
563 C
564 C
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C
580 C
581 C
582 C
583 C
584 C
585 C
586 C
587 C
588 C
589 C
590 C
591 C
592 C
593 C
594 C
595 C
596 C
597 C
598 C
599 C
600 C
601 C
602 C
603 C
604 C
605 C
606 C
607 C
608 C
609 C
610 C
611 C
612 C
613 C
614 C
615 C
616 C
617 C
618 C
619 C
620 C
621 C
622 C
623 C
624 C
625 C
626 C
627 C
628 C
629 C
630 C
631 C
632 C
633 C
634 C
635 C
636 C
637 C
638 C
639 C
640 C
641 C
642 C
643 C
644 C
645 C
646 C
647 C
648 C
649 C
650 C
651 C
652 C
653 C
654 C
655 C
656 C
657 C
658 C
659 C
660 C
661 C
662 C
663 C
664 C
665 C
666 C
667 C
668 C
669 C
670 C
671 C
672 C
673 C
674 C
675 C
676 C
677 C
678 C
679 C
680 C
681 C
682 C
683 C
684 C
685 C
686 C
687 C
688 C
689 C
690 C
691 C
692 C
693 C
694 C
695 C
696 C
697 C
698 C
699 C
700 C
701 C
702 C
703 C
704 C
705 C
706 C
707 C
708 C
709 C
710 C
711 C
712 C
713 C
714 C
715 C
716 C
717 C
718 C
719 C
720 C
721 C
722 C
723 C
724 C
725 C
726 C
727 C
728 C
729 C
730 C
731 C
732 C
733 C
734 C
735 C
736 C
737 C
738 C
739 C
740 C
741 C
742 C
743 C
744 C
745 C
746 C
747 C
748 C
749 C
750 C
751 C
752 C
753 C
754 C
755 C
756 C
757 C
758 C
759 C
760 C
761 C
762 C
763 C
764 C
765 C
766 C
767 C
768 C
769 C
770 C
771 C
772 C
773 C
774 C
775 C
776 C
777 C
778 C
779 C
780 C
781 C
782 C
783 C
784 C
785 C
786 C
787 C
788 C
789 C
790 C
791 C
792 C
793 C
794 C
795 C
796 C
797 C
798 C
799 C
800 C
801 C
802 C
803 C
804 C
805 C
806 C
807 C
808 C
809 C
810 C
811 C
812 C
813 C
814 C
815 C
816 C
817 C
818 C
819 C
820 C
821 C
822 C
823 C
824 C
825 C
826 C
827 C
828 C
829 C
830 C
831 C
832 C
833 C
834 C
835 C
836 C
837 C
838 C
839 C
840 C
841 C
842 C
843 C
844 C
845 C
846 C
847 C
848 C
849 C
850 C
851 C
852 C
853 C
854 C
855 C
856 C
857 C
858 C
859 C
860 C
861 C
862 C
863 C
864 C
865 C
866 C
867 C
868 C
869 C
870 C
871 C
872 C
873 C
874 C
875 C
876 C
877 C
878 C
879 C
880 C
881 C
882 C
883 C
884 C
885 C
886 C
887 C
888 C
889 C
890 C
891 C
892 C
893 C
894 C
895 C
896 C
897 C
898 C
899 C
900 C
901 C
902 C
903 C
904 C
905 C
906 C
907 C
908 C
909 C
910 C
911 C
912 C
913 C
914 C
915 C
916 C
917 C
918 C
919 C
920 C
921 C
922 C
923 C
924 C
925 C
926 C
927 C
928 C
929 C
930 C
931 C
932 C
933 C
934 C
935 C
936 C
937 C
938 C
939 C
940 C
941 C
942 C
943 C
944 C
945 C
946 C
947 C
948 C
949 C
950 C
951 C
952 C
953 C
954 C
955 C
956 C
957 C
958 C
959 C
960 C
961 C
962 C
963 C
964 C
965 C
966 C
967 C
968 C
969 C
970 C
971 C
972 C
973 C
974 C
975 C
976 C
977 C
978 C
979 C
980 C
981 C
982 C
983 C
984 C
985 C
986 C
987 C
988 C
989 C
990 C
991 C
992 C
993 C
994 C
995 C
996 C
997 C
998 C
999 C
1000 C

```



```

301 GO TO 350
302
303 C
304 C SWITCH NO.3 Remove last point digitized
305
306 540 IF (N .LT. 1) GO TO 550
307 GO TO 560
308 550 WRITE (6,600)
309 GO TO 350
310 560 N = N - 1
311 ICK = ICKP
312 ICKP = ICK
313 GO TO 1350
314
315 C
316 C SWITCH NO.4 Remove last Transmissivity zone Generated
317 C
318 570 ITRNS = ITRNS - 1
319 NPTS = NPTSP
320 NPTS1 = NPTSP + 1
321 GO TO 350
322
323 C
324 C SWITCH NO.6 "Pic" Constant Head Nodes
325 C
326 580 IF (N .LT. 1) GO TO 590
327 GO TO 610
328 590 WRITE (6,600)
329 600 FORMAT ('YOU HAVE NOT YET GENERATED A MESH')
330 GO TO 350
331 610 IF (ID .GT. 0) GO TO 690
332 WRITE (6,620)
333 620 FORMAT ('MOVE THE CURSOR WITH THE JOYSTICK TO INDICATE THE',/,
334 1 'CONSTANT HEAD NODES IN THIS VIEW. TOUCH A DIFFERENT KEY',
335 2 ', /, 'TO ENTER EACH DIFFERENT CONSTANT HEAD NODE.')
336 ICON = 0
337 C *****
338 630 CALL IGCRTL('TERMINAL', 'ERASE')
339 C *****
340 CALL IGINIT
341 C *****
342 CALL IGBGNS('PLOT')
343 C *****
344 CALL IGMA(O.O, O.O)
345 DO 650 LL = 1, NELM
346 JJ = IN(LL,1)
347 XO = (XN(JJ) - XC) / XC
348 YO = (YN(JJ) - YC) / YC
349 CALL IGMA(XO, YO)
350 C *****
351 DO 640 L = 1, INC

```



```

351      JJ = IN(LL,L)
352      XO = (XN(JJ) - XC) / XC
353      YO = (YN(JJ) - YC) / YC
354      C *****
355      CALL IGDA(XO, YO)
356      C *****
357      640 CONTINUE
358      JJ = IN(LL,1)
359      XO = (XN(JJ) - XC) / XC
360      YO = (YN(JJ) - YC) / YC
361      C *****
362      CALL IGDA(XO, YO)
363      C *****
364      650 CONTINUE
365      C *****
366      CALL IGENDS('PLOT')
367      C *****
368      CALL IGDRON('TERMINAL')
369      C *****
370      CALL IGXYIN(XIN(IC), YIN(IC))
371      C *****
372      XP = XIN(IC) * XC + XC
373      YP = YIN(IC) * YC + YC
374      DO 670 I = 1, NPTS
375      IF (XN(I) .LT. XC - 0.1*SCL .OR. XN(I) .GT. XC + 0.1*SCL)
376      1 GO TO 670
377      IF (YN(I) .LT. YP - 0.1*SCL .OR. YN(I) .GT. YP + 0.1*SCL)
378      1 GO TO 670
379      ITYPE(I) = 1
380      WRITE (6,660)
381      FORMAT ('More Constant head nodes to be defined?')
382      READ (5,1610) IANSWR
383      IF (IANSWR .EQ. YES) GO TO 630
384      C *****
385      CALL IGCTRL('TERMINAL', 'ERASE')
386      C *****
387      GO TO 350
388      670 CONTINUE
389      WRITE (6,680)
390      FORMAT ('MISSED, TRY AGAIN')
391      GO TO 630
392      690 WRITE (6,700)
393      700 FORMAT ('This routine will establish constant head nodes',/,
394      1 'for each side of the generated mesh in the sequence',/,
395      2 'in which they are queried. The constant head value',/,
396      3 'will be equal to the elevation of each node',/,
397      4 'Top face?')
398      READ (5,1610) IANSWR
399      IF (IANSWR .EQ. NO) GO TO 730
400      ICO = IFE(1)

```



```

401 DO 720 IC01 = 1, IC0
402 DO 710 NC = 1, 4
403     ITYPE(IN(IFACE(IC01,1),NC)) = 1
404
405     710 CONTINUE
406
407     720 CONTINUE
408
409     730 WRITE (6,740)
410     740 FORMAT ('Bottom face?')
411     READ (5,1610) IANSWR
412     IF (IANSWR.EQ. NO) GO TO 770
413     IC0 = IFE(2)
414
415     DO 760 IC01 = 1, IC0
416     DO 750 NC = 5, 8
417         ITYPE(IN(IFACE(IC01,2),NC)) = 1
418
419     750 CONTINUE
420
421     760 CONTINUE
422
423     770 WRITE (6,780)
424     780 FORMAT ('South face?')
425     READ (5,1610) IANSWR
426     IF (IANSWR.EQ. NO) GO TO 800
427     IC0 = IFE(3)
428
429     DO 790 IC01 = 1, IC0
430     ITYPE(IN(IFACE(IC01,3),2)) = 1
431     ITYPE(IN(IFACE(IC01,3),6)) = 1
432     ITYPE(IN(IFACE(IC01,3),7)) = 1
433     ITYPE(IN(IFACE(IC01,3),3)) = 1
434
435     790 CONTINUE
436
437     800 WRITE (6,810)
438     810 FORMAT ('East face?')
439     READ (5,1610) IANSWR
440     IF (IANSWR.EQ. NO) GO TO 830
441     IC0 = IFE(4)
442
443     DO 820 IC01 = 1, IC0
444     ITYPE(IN(IFACE(IC01,4),3)) = 1
445     ITYPE(IN(IFACE(IC01,4),7)) = 1
446     ITYPE(IN(IFACE(IC01,4),8)) = 1
447     ITYPE(IN(IFACE(IC01,4),4)) = 1
448
449     820 CONTINUE
450
451     830 WRITE (6,840)
452     840 FORMAT ('North face?')
453     READ (5,1610) IANSWR
454     IF (IANSWR.EQ. NO) GO TO 860
455     IC0 = IFE(5)
456
457     DO 850 IC01 = 1, IC0
458     ITYPE(IN(IFACE(IC01,5),4)) = 1
459     ITYPE(IN(IFACE(IC01,5),8)) = 1
460     ITYPE(IN(IFACE(IC01,5),5)) = 1
461     ITYPE(IN(IFACE(IC01,5),1)) = 1
462
463     850 CONTINUE
464
465     860 WRITE (6,870)
466     870 FORMAT ('West face?')

```



```

451 READ (5,1610) IANSWR
452 IF (IANSWR .EQ. NO) GO TO 890
453 ICO = IFE(6)
454 DO 880 ICO1 = 1, ICO
455   ITYPE(IN(IFACE(ICO1,6),1)) = 1
456   ITYPE(IN(IFACE(ICO1,6),5)) = 1
457   ITYPE(IN(IFACE(ICO1,6),6)) = 1
458   ITYPE(IN(IFACE(ICO1,6),2)) = 1
459 880 CONTINUE
460 C Count the number of constant head nodes
461 890 DO 900 JCON = 1, NPTS
462   IF (ITYPE(JCON) .NE. 1) GO TO 900
463   ICON = JCON + 1
464 900 CONTINUE
465   GO TO 350
466
467 C
468 C
469 C SWITCH NO.5 DIGITIZE LEAKANCE ELEMENTS
470
471 910 IF (N .LT. 1) GO TO 920
472   GO TO 930
473 920 WRITE (6,600)
474   GO TO 350
475 930 WRITE (6,940)
476 940 FORMAT ('Move the cursor with the joystick to indicate the',/,
477   1 'leakance elements in this view. Youch a different key to'
478   2 ' , /, 'to enter a different leakance element. ')
479 950 WRITE (6,960)
480 960 FORMAT ('*****Enter leakance value*****')
481   NLN = NLN + 1
482   READ (5,970) ALEAK(NLN)
483 970 FORMAT (1E10.4)
484 C *****
485 CALL IGCRTL('TERMINAL', 'ERASE')
486 C *****
487 CALL IGINIT
488 C *****
489 CALL IGBGNS('PLOT')
490 C *****
491 CALL IGMA(O.O, O.O)
492 C *****
493 DO 990 LL = 1, NELM
494   JJ = IN(LL,1)
495   XO = (XN(JJ) - XC) / XC
496   YO = (YN(JJ) - YC) / YC
497 C *****
498 CALL IGMA(XO, YO)
499 C *****
500 DO 980 L = 1, INC
   JJ = IN(LL,L)
   XO = (XN(JJ) - XC) / XC

```



```

501      YO = (YN(JJ) - YC) / YC
502      C *****
503      CALL IGDA(XO, YO)
504      C *****
505      980 CONTINUE
506      JJ = IN(LL,1)
507      XO = (XN(JJ) - XC) / XC
508      YO = (YN(JJ) - YC) / YC
509      C *****
510      CALL IGDA(XO, YO)
511      C *****
512      990 CONTINUE
513      C *****
514      CALL IGENDS('PLOT')
515      C *****
516      CALL IGDRON('TERMINAL')
517      C *****
518      CALL IGXYIN(XIN(IL), YIN(IL))
519      C *****
520      XP = XIN(IL) * XC + XC
521      YP = YIN(IL) * YC + YC
522      DO 1020 J = 1, NELM
523          X = O.
524          Y = O.
525          DO 1000 K = 1, INC
526              X = XN(IN(J,K)) + X
527              Y = YN(IN(J,K)) + Y
528          1000 CONTINUE
529          XMEAN = X / INC
530          YMEAN = Y / INC
531          IF (XP .LT. XMEAN - 0.1*SCL .OR. XP .GT. XMEAN + 0.1*SCL)
532              1 GO TO 1020
533          IF (YP .LT. YMEAN - 0.1*SCL .OR. YP .GT. YMEAN + 0.1*SCL)
534              1 GO TO 1020
535          LEAK(NLN) = J
536          WRITE (6,1010)
537          FORMAT ('More leakance elements to be defined?')
538          READ (5,1610) IANSWR
539          IF (IANSWR .EQ. YES) GO TO 950
540      C *****
541      CALL IGCTRL('TERMINAL', 'ERASE')
542      C *****
543      GO TO 350
544      1020 CONTINUE
545      WRITE (6,680)
546      GO TO 950
547      C
548      C SWITCH NO.7 DIGITIZE A PUMPING WELL IN X-Y PLANE
549      C
550      1030 WRITE (6,1040)

```



```

1040 FORMAT ('The location of a pumping well in the x-y plane will be',
1      /, 'defined. If more than one pumping well exists, select
2switch', /, 'No. 7 again.', /, 'Enter the following pump informati
3on', /)
IP = IP + 1
IF (ID .GT. 0) GO TO 1100
1050 IPL = IPL + 1
WRITE (6,1060)
1060 FORMAT ('Enter the start time:')
READ (5,120) TIME(IPL,1)
WRITE (6,1070)
1070 FORMAT ('Enter the finish time:')
READ (5,120) TIME(IPL,2)
WRITE (6,1080)
1080 FORMAT ('Enter the number of time steps:')
READ (5,1090) NTIME(IPL)
1090 FORMAT (I4)
GO TO 1150
1100 IPL = IPL + 1
WRITE (6,1110)
1110 FORMAT ('Enter the simulation period:')
READ (5,120) TIME(IPL,2)
WRITE (6,1120)
1120 FORMAT ('Enter the initial time step:')
READ (5,120) TIME(IPL,1)
WRITE (6,1130)
1130 FORMAT ('Enter the multiplier for increasing the time step:')
READ (5,120) CHNG(IPL)
WRITE (6,1140)
1140 FORMAT ('Enter the number of time steps between', /,
1      /, 'changes in delta time:')
READ (5,1090) ITCHNG(IPL)
1150 WRITE (6,1160)
1160 FORMAT ('Enter the pumping rate in cu.ft./sec:')
READ (5,120) PMPRTE(IPL,IP)
WRITE (6,1170)
1170 FORMAT ('More pumping information?')
READ (5,1610) IANSWR
IF (ID .GT. 0 .AND. IANSWR .EQ. YES) GO TO 1100
IF (IANSWR .EQ. YES) GO TO 1050
WRITE (6,1340)
READ (5,390,END=2260) IX, IY
IF (IX .GT. 1000) GO TO 430
1180 IF (IP .GT. 9) GO TO 470
PPOSX(IP) = X
PPOSY(IP) = Y
JCHAR = 92
GO TO 350

```

C

C

```

551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600

```



```

601 C SWITCH NO. 8 DIGITIZE OBSERVATION WELL IN X-Y PLANE
602 C
603 1190 WRITE (6,1200)
604 1200 FORMAT ('THE LOCATION OF AN OBSERVATION WELL IN THE X-Y PLANE ', /
605 1
606 260 'WILL BE DEFINED. IF MORE THAN ONE PUMPING WELL EXISTS, SELECT', /
607 1,
608 2'SWITCH NO. 8 AGAIN.')
609 IO = IO + 1
610 READ (5,390,END=2260) IX, IY
611 IF (IX .GT. 1000) GO TO 430
612 1220 IF (IP .GT. 9) GO TO 470
613 OBSX(IO) = X
614 OBSY(IO) = Y
615 JCHAR = 92
616 GO TO 350
617
618
619 C SWITCH NO. 9 DIGITIZE TRANSMISSIVITY BOUNDARY X-Y PLANE
620 C
621 1230 WRITE (6,1240)
622 1240 FORMAT ('Digitize the location of the transmissivity boundaries',
623 1 /,
624 260 'in the x-y plane. It is necessary that you digitize around the',
625 1 /,
626 490 'entire perimeter of the transmissivity zone so that the program',
627 1 /, 'will be able to generate the mesh within this boundary. If',
628 2 /, 'the user is working in 3-D, it is necessary to digitize the',
629 3, /, 'perimeter of the upper surface of the zone after the lower',
630 4 /,
631 1250 'surface is done. If there is more than one transmissivity zone,'
632 1, /, 'select switch No.9 again after the mesh for the previous zon
633 2e', /, 'has been generated.', /, '*****ENTER THE STORATIVITY*****'
634 3, / )
635 ICKP = ICK
636 NCEL(ITRNS,2) = NELM
637 ITRNS = ITRNS + 1
638 NCEL(ITRNS,1) = NELM1
639 NPTSP = NPTS
640 NPTSP1 = NPTS + 1
641 N = NPTS1
642 READ (5,1260) STOR(ITRNS)
643 1260 FORMAT (E10.3)
644 IF (ID .GT. 0) GO TO 1280
645 WRITE (6,1270)
646 1270 FORMAT ('Enter the transmissivity')
647 READ (5,1260) COX(ITRNS)
648 GO TO 1330
649 1280 WRITE (6,1320)
650 WRITE (6,1290)

```



```

651 1290 FORMAT ('Enter the hydraulic conductivity in the x-direction')
652 READ (5,1260) COX(ITRNS)
653 WRITE (6,1300)
654 1300 FORMAT ('Enter the hydraulic conductivity in the y-direction')
655 READ (5,1260) COY(ITRNS)
656 WRITE (6,1310)
657 READ (5,1260) COZ(ITRNS)
658 1310 FORMAT ('Enter the hydraulic conductivity in the z-direction')
659 1320 FORMAT ('Digitize the x-y coordinates of the upper surface in the'
660 1 , /, 'same manner as the lower surface has been digitized fi
661 2rst.')
```

```

662 1330 WRITE (6,1340)
663 1340 FORMAT ('NOW BEGIN DIGITIZING')
664 1350 READ (5,390,END=2260) IX, IY
665 IF (IX .GT. 1000) GO TO 430
666 ICK = IY / 1000 + 1
667 WRITE (6,400) ICK
668 IF (ICK .GT. 13) GO TO 350
669 GO TO (350, 510, 540, 570, 910, 580, 1030, 1190, 1230, 1370,
670 11380, 1590, 2000), ICK
671 1360 IF (N .GT. 4998) GO TO 470
672 XN(N) = X
673 YN(N) = Y
674 N = N + 1
675 NPTS = N - 1
676 GO TO 1350
677
678 C SWITCH NO. 10 Future Use
679 C
680 1370 GO TO 350
681 C
682 C Switch No.11 Input Boundary Fluxes
683 C
684 1380 IF (ID .GT. 0) GO TO 1400
685 WRITE (6,1390)
686 1390 FORMAT ('This switch has no significance in 2-D')
687 GO TO 350
688 1400 WRITE (6,1410)
689 1410 FORMAT ('This routine has the capability to give an entire', /,
690 1 'face of a generated mesh a boundary flux value. The', /,
691 2 'user will be queried as to whether input a boundary', /,
692 3 'flux for each face queried.', /, 'Top face?')
693 IB = 0
694 READ (5,1610) IANSWR
695 IF (IANSWR .EQ. N0) GO TO 1440
696 IB = IB + 1
697 READ (5,1420) BFL
698 BFL = BFL / (IFE(1)*4.)
699 1420 FORMAT (F10.0)
700 ICO = IFE(1)
```



```

701 DO 1430 ICO1 = 1, ICO
702     ITYPE(IN(IFACE(ICO1,1),1)) = 2
703     BFLUX(IN(IFACE(ICO1,1),1)) = BFLUX(IN(IFACE(ICO1,1),1)) + BFL
704     ITYPE(IN(IFACE(ICO1,1),2)) = 2
705     BFLUX(IN(IFACE(ICO1,1),2)) = BFLUX(IN(IFACE(ICO1,1),2)) + BFL
706     ITYPE(IN(IFACE(ICO1,1),3)) = 2
707     BFLUX(IN(IFACE(ICO1,1),3)) = BFLUX(IN(IFACE(ICO1,1),3)) + BFL
708     ITYPE(IN(IFACE(ICO1,1),4)) = 2
709     BFLUX(IN(IFACE(ICO1,1),4)) = BFLUX(IN(IFACE(ICO1,1),4)) + BFL
710
711 1430 CONTINUE
712 1440 WRITE (6,1450)
713 1450 FORMAT ('Bottom face?')
714     READ (5,1610) IANSWR
715     IF (IANSWR .EQ. NO) GO TO 1470
716     READ (5,1420) BFL
717     BFL = BFL / (IFE(2)*4.)
718     ICO = IFE(2)
719 DO 1460 ICO1 = 1, ICO
720     ITYPE(IN(IFACE(ICO1,2),5)) = 2
721     BFLUX(IN(IFACE(ICO1,2),5)) = BFLUX(IN(IFACE(ICO1,2),5)) + BFL
722     ITYPE(IN(IFACE(ICO1,2),6)) = 2
723     BFLUX(IN(IFACE(ICO1,2),6)) = BFLUX(IN(IFACE(ICO1,2),6)) + BFL
724     ITYPE(IN(IFACE(ICO1,2),7)) = 2
725     BFLUX(IN(IFACE(ICO1,2),7)) = BFLUX(IN(IFACE(ICO1,2),7)) + BFL
726     ITYPE(IN(IFACE(ICO1,2),8)) = 2
727     BFLUX(IN(IFACE(ICO1,2),8)) = BFLUX(IN(IFACE(ICO1,2),8)) + BFL
728
729 1460 CONTINUE
730 1470 WRITE (6,1480)
731 1480 FORMAT ('South face?')
732     READ (5,1610) IANSWR
733     IF (IANSWR .EQ. NO) GO TO 1500
734     READ (5,1420) BFL
735     BFL = BFL / (IFE(3)*4.)
736     ICO = IFE(3)
737 DO 1490 ICO1 = 1, ICO
738     ITYPE(IN(IFACE(ICO1,3),2)) = 2
739     BFLUX(IN(IFACE(ICO1,3),2)) = BFLUX(IN(IFACE(ICO1,3),2)) + BFL
740     ITYPE(IN(IFACE(ICO1,3),6)) = 2
741     BFLUX(IN(IFACE(ICO1,3),6)) = BFLUX(IN(IFACE(ICO1,3),6)) + BFL
742     ITYPE(IN(IFACE(ICO1,3),7)) = 2
743     BFLUX(IN(IFACE(ICO1,3),7)) = BFLUX(IN(IFACE(ICO1,3),7)) + BFL
744     ITYPE(IN(IFACE(ICO1,3),3)) = 2
745     BFLUX(IN(IFACE(ICO1,3),3)) = BFLUX(IN(IFACE(ICO1,3),3)) + BFL
746
747 1490 CONTINUE
748 1500 WRITE (6,1510)
749 1510 FORMAT ('East face?')
750     READ (5,1610) IANSWR
751     IF (IANSWR .EQ. NO) GO TO 1530
752     READ (5,1420) BFL
753     ICO = IFE(4)
754     READ (5,1420) BFL

```



```

751 BFL = BFL / (IFE(4)*4.)
752 DO 1520 ICO1 = 1, ICO
753 ITYPE(IN(IFACE(ICO1,4),3)) = 2
754 BFLUX(IN(IFACE(ICO1,4),3)) = BFLUX(IN(IFACE(ICO1,4),3)) + BFL
755 ITYPE(IN(IFACE(ICO1,4),7)) = 2
756 BFLUX(IN(IFACE(ICO1,4),7)) = BFLUX(IN(IFACE(ICO1,4),7)) + BFL
757 ITYPE(IN(IFACE(ICO1,4),8)) = 2
758 BFLUX(IN(IFACE(ICO1,4),8)) = BFLUX(IN(IFACE(ICO1,4),8)) + BFL
759 ITYPE(IN(IFACE(ICO1,4),4)) = 2
760 BFLUX(IN(IFACE(ICO1,4),4)) = BFLUX(IN(IFACE(ICO1,4),4)) + BFL
761
1520 CONTINUE
762
1530 WRITE (6,1540)
763
1540 FORMAT ('North face?')
764
765 READ (5,1610) IANSWR
766 IF (IANSWR.EQ. NO) GO TO 1560
767 ICO = IFE(5)
768 READ (5,1420) BFL
769 BFL = BFL / (IFE(5)*4.)
770 DO 1550 ICO1 = 1, ICO
771 ITYPE(IN(IFACE(ICO1,5),4)) = 2
772 BFLUX(IN(IFACE(ICO1,5),4)) = BFLUX(IN(IFACE(ICO1,5),4)) + BFL
773 ITYPE(IN(IFACE(ICO1,5),8)) = 2
774 BFLUX(IN(IFACE(ICO1,5),8)) = BFLUX(IN(IFACE(ICO1,5),8)) + BFL
775 ITYPE(IN(IFACE(ICO1,5),5)) = 2
776 BFLUX(IN(IFACE(ICO1,5),5)) = BFLUX(IN(IFACE(ICO1,5),5)) + BFL
777 ITYPE(IN(IFACE(ICO1,5),1)) = 2
778 BFLUX(IN(IFACE(ICO1,5),1)) = BFLUX(IN(IFACE(ICO1,5),1)) + BFL
779
1550 CONTINUE
780
1560 WRITE (6,1570)
781
1570 FORMAT ('west face?')
782
783 READ (5,1610) IANSWR
784 IF (IANSWR.EQ. NO) GO TO 350
785 ICO = IFE(6)
786 READ (5,1420) BFL
787 BFL = BFL / (IFE(6)*4.)
788 DO 1580 ICO1 = 1, ICO
789 ITYPE(IN(IFACE(ICO1,6),1)) = 2
790 BFLUX(IN(IFACE(ICO1,6),1)) = BFLUX(IN(IFACE(ICO1,6),1)) + BFL
791 ITYPE(IN(IFACE(ICO1,6),5)) = 2
792 BFLUX(IN(IFACE(ICO1,6),5)) = BFLUX(IN(IFACE(ICO1,6),5)) + BFL
793 ITYPE(IN(IFACE(ICO1,6),6)) = 2
794 BFLUX(IN(IFACE(ICO1,6),6)) = BFLUX(IN(IFACE(ICO1,6),6)) + BFL
795 ITYPE(IN(IFACE(ICO1,6),2)) = 2
796 BFLUX(IN(IFACE(ICO1,6),2)) = BFLUX(IN(IFACE(ICO1,6),2)) + BFL
797
1580 CONTINUE
798 GO TO 350
799
C
C SWITCH NO. 12 STORE INFO FOR LATER USE
C
1590 WRITE (6,1600)
800

```



```

801 1600 FORMAT ('DO YOU WISH TO STOP AND STORE THE DATA ?')
802 READ (5,1610) MQUES6
803 1610 FORMAT (A1)
804 IF (MQUES6.EQ. NO) GO TO 1630
805 NCEL(ITRNS,2) = NELM
806 WRITE (6,1620)
807 1620 FORMAT ('Device 7 contains digital info., Device 9 contains plot',
808 1 /, 'data: Store appropriate files attached to these device
809 2s.')
810 GO TO 1650
811 1630 WRITE (6,1640)
812 1640 FORMAT ('THE PLOT WAS NOT ENDED, SELECT A NEW SWITCH')
813 GO TO 350
814 C *****
815 1650 CALL SETUP(IDIFF)
816 C *****
817 CALL OPTNUM(IDIFF, NBW)
818 C *****
819 IF (ID.GT. 0) GO TO 1830
820 WRITE (7,1660)
821 1660 FORMAT ('FE')
822 WRITE (7,1670) NPTS, NELM, NBW, NLN
823 1670 FORMAT (4I5)
824 DO 1700 I = 1, NPTS
825 WRITE (7,1680) I, XN(I), YN(I), ITYPE(I), CX(I)
826 1680 FORMAT (I5, 2F10.3, I5, F10.3)
827 1690 FORMAT (I5, 2F10.3, 4X, 'C', F10.4)
828 1700 CONTINUE
829 DO 1740 J = 1, NELM
830 IF (INC.EQ. 3) WRITE (7,1710) (J, (IN(J,K), K=1,3))
831 IF (INC.EQ. 4) WRITE (7,1720) (J, (IN(J,K), K=1,4))
832 IF (INC.EQ. 8) WRITE (7,1730) (J, (IN(J,K), K=1,8))
833 1710 FORMAT (4I5)
834 1720 FORMAT (5I5)
835 1730 FORMAT (9I5)
836 1740 CONTINUE
837 DO 1760 I = 1, NLN
838 WRITE (7,1750) LEAK(I), ALEAK(I)
839 1750 FORMAT (I5, 1E10.4)
840 1760 CONTINUE
841 WRITE (7,1770) STORAT
842 1770 FORMAT (E10.3)
843 DO 1810 I = 1, IPL
844 WRITE (7,1780) TIME(I,1), TIME(I,2), NTIME(I), IP
845 1780 FORMAT (2F10.5, 2I5)
846 DO 1800 I1 = 1, IP
847 WRITE (7,1790) I1, I1, PMPRTE(I,I1)
848 1790 FORMAT ('PW', I2, I6, F10.7)
849 1800 CONTINUE
850 1810 CONTINUE

```



```

851 WRITE (7,1820)
852 FORMAT (/, 'XX')
853 GO TO 350
854 WRITE (7,1880)
855 WRITE (7,1890) TITLE
856 WRITE (7,1900) NPTS, NELM, NBW, ICON, ITRNS
857 WRITE (7,1910) (I,XN(I),YN(I),ZN(I),ITYPE(I),I=1,NPTS)
858 WRITE (7,1920) (J,(IN(J,INN(L)),L=1,INC),J=1,NELM)
859 WRITE (7,1940) BETA, DENS
860 DO 1840 IT = 1, ITRNS
861 WRITE (7,1930) NCEL(IT,1), NCEL(IT,2), COX(IT), COY(IT),
862 COZ(IT), HETA, TETA, AL
863 1 COZ(IT), HETA, TETA, AL
864 CONTINUE
865 ISPL = 0
866 WRITE (7,1950) ISPL
867 IF (ISS.LT. 1) GO TO 1870
868 DO 1860 IPU = 1, IP
869 WRITE (7,1960) TIME(IPU,2), TIME(IPU,1), CHNG(IPU), ITCHNG(IPU),
870 1 IPPN(IPU)
871 IPNN = IPPN(IPU)
872 DO 1850 IPN1 = 1, IPNN
873 WRITE (7,1980) NPMP(IPN1), PMPRTE(IPN1,IPU)
874 CONTINUE
875 1850 CONTINUE
876 1860 CONTINUE
877 WRITE (7,1970)
878 WRITE (7,1990) LINEAR')
879 1880 FORMAT ('ISOB
880 1890 FORMAT (20A4)
881 1900 FORMAT (5I5)
882 1910 FORMAT (I5, 3F15.3, I5)
883 1920 FORMAT (9I5)
884 1930 FORMAT (2I5, 7E10.3)
885 1940 FORMAT (2F15.12)
886 1950 FORMAT (I5)
887 1960 FORMAT (3F10.2, 2I10)
888 1970 FORMAT (6X, '999.')
889 1980 FORMAT (I5, F10.5)
890 1990 FORMAT (/, 'XXXX')
891 GO TO 350
892 C
893 C SWITCH NO.13
894 C
895 2000 WRITE (6,2010)
896 2010 FORMAT ('DO YOU WISH TO GENERATE A MESH? (Y OR N)')
897 READ (5,1610) MQUES7
898 IF (MQUES7.EQ. NO) GO TO 350
899 YTOP = AGSENS('TERMINAL','YSIZE')
900 CALL IGRAN('*MP*', 'SCALE', YTOP)
901 IZD = 0
902 INC = 4

```



```

901 IF (IELEM.EQ. 0 .AND. ID.LT. 1) INC = 3
902 IF (IELEM.EQ. 1 .AND. ID.EQ. 1) INC = 8
903 C *****
904 CALL DSTRB
905 C *****
906 IF (IP.GT. 0) CALL SMOOTH
907 NCEL(ITRNS,2) = NELM
908 C *****
909 IF (ID.GT. 0) GO TO 2060
910 2020 CALL IGINIT
911 C *****
912 C *****
913 C *****
914 DO 2050 ITR = 1, ITRNS
915 NE1 = NCEL(ITR,1)
916 NE2 = NCEL(ITR,2)
917 CALL IGMA(O.O, O.O)
918 C *****
919 DO 2040 LL = NE1, NE2
920 IPIC = IGBGNS(O)
921 XLIMIT = 1.0
922 JJ = IN(LL,1)
923 XO = (XN(JJ) - XC) / XC
924 YO = ((YN(JJ) - YC)/YC)
925 C *****
926 CALL IGMA(XO, YO)
927 C *****
928 DO 2030 L = 1, INC
929 JJ = IN(LL,L)
930 XO = (XN(JJ) - XC) / XC
931 YO = ((YN(JJ) - YC)/YC)
932 C *****
933 CALL IGDA(XO, YO)
934 C *****
935 2030 CONTINUE
936 JJ = IN(LL,1)
937 XO = (XN(JJ) - XC) / XC
938 YO = ((YN(JJ) - YC)/YC)
939 C *****
940 CALL IGDA(XO, YO)
941 C *****
942 2040 CONTINUE
943 C *****
944 CALL IGHUE(IPIC, COL(ITR))
945 C *****
946 CALL IGENDS(IPIC)
947 2050 CONTINUE
948 C *****
949 CALL IGDRON('TERMINAL')
950 C *****

```



```

951 CALL ASKPLT
952 C *****
953 CALL IGCTRL('TERMINAL', 'ERASE')
954 C *****
955 NPTS1 = NPTS + 1
956 NELM1 = NELM + 1
957 IF (ID.LT. 1) GO TO 350
958 2060 CONTINUE
959 IF (IELEM.EQ. 1) GO TO 2100
960 ZO = O.O
961 C *****
962 CALL IGINIT
963 C *****
964 CALL IGCTRL('TERMINAL', 'ERASE')
965 C *****
966 CALL IGBGNS('PICT')
967 C *****
968 DO 2160 ITR = 1, ITRNS
969 NE1 = NCEL(ITR,1)
970 NE2 = NCEL(ITR,2)
971 CALL IGMA(O.O, O.O, O.O)
972 C *****
973 DO 2090 LL = NE1, NE2
974 XLIMIT = 1.O
975 JJ = IN(LL,1)
976 XO = (XN(JJ) - XC) / XC
977 YO = ((YN(JJ) - YC)/YC)
978 ZO = ZN(JJ) / ZC + 1.
979 C *****
980 CALL IGMA(XO, YO, ZO)
981 C *****
982 DO 2070 L = 1, INC
983 JJ = IN(LL,L)
984 XO = (XN(JJ) - XC) / XC
985 YO = ((YN(JJ) - YC)/YC)
986 ZO = ZN(JJ) / ZC + 1.
987 C *****
988 CALL IGDA(XO, YO, ZO)
989 C *****
990 2070 CONTINUE
991 JJ = IN(LL,1)
992 XO = (XN(JJ) - XC) / XC
993 YO = ((YN(JJ) - YC)/YC)
994 ZO = ZN(JJ) / ZC + 1.
995 2080 CONTINUE
996 C *****
997 CALL IGDA(XO, YO, ZO)
998 C *****
999 2090 CONTINUE
1000 GO TO 2150

```



```

1001 C *****
1002 2100 CALL IGINIT
1003      ZO=0.0
1004 C *****
1005     CALL IGCRL('TERMINAL', 'ERASE')
1006 C *****
1007     CALL IGBGNS('PICT')
1008 C *****
1009     CALL IGMA(O.O, O.O, O.O)
1010 C *****
1011     DO 2140 LL = NE1, NE2
1012         JJ = IN(LL,1)
1013         XO = (XN(JJ) - XC) / XC
1014         YO = ((YN(JJ) - YC)/YC)
1015         ZO = ZN(JJ)/ZC+1
1016 C *****
1017     CALL IGMA(XO, YO, ZO)
1018 C *****
1019     DO 2120 L = 1, 3
1020         X1(1) = (XN(IN(LL,L)) - XC) / XC
1021         Y1(1) = ((YN(IN(LL,L)) - YC)/YC)
1022         Z1(1) = (ZN(IN(LL,L)) - ZC) / ZC + 1.
1023         X1(2) = (XN(IN(LL,L + 4)) - XC) / XC
1024         Y1(2) = ((YN(IN(LL,L + 4)) - YC)/YC)
1025         Z1(2) = (ZN(IN(LL,L + 4)) - ZC) / ZC + 1.
1026         X1(3) = (XN(IN(LL,L + 5)) - XC) / XC
1027         Y1(3) = ((YN(IN(LL,L + 5)) - YC)/YC)
1028         Z1(3) = (ZN(IN(LL,L + 5)) - ZC) / ZC + 1.
1029         X1(4) = (XN(IN(LL,L + 1)) - XC) / XC
1030         Y1(4) = ((YN(IN(LL,L + 1)) - YC)/YC)
1031         Z1(4) = (ZN(IN(LL,L + 1)) - ZC) / ZC + 1.
1032 C *****
1033     CALL IGMA(X1(1), Y1(1), Z1(1))
1034 C *****
1035     DO 2110 L1 = 1, 4
1036 C *****
1037         CALL IGDA(X1(L1), Y1(L1), Z1(L1))
1038 C *****
1039         CONTINUE
1040 C *****
1041         CALL IGDA(X1(1), Y1(1), Z1(1))
1042 C *****
1043         CONTINUE
1044         X1(1) = (XN(IN(LL,1)) - XC) / XC
1045         Y1(1) = ((YN(IN(LL,1)) - YC)/YC)
1046         Z1(1) = (ZN(IN(LL,1)) - ZC) / ZC + 1.
1047         X1(2) = (XN(IN(LL,5)) - XC) / XC
1048         Y1(2) = ((YN(IN(LL,5)) - YC)/YC)
1049         Z1(2) = (ZN(IN(LL,5)) - ZC) / ZC + 1.
1050         X1(3) = (XN(IN(LL,8)) - XC) / XC

```



```

1051 Y1(3) = ((YN(IN(LL,8)) - YC)/YC)
1052 Z1(3) = (ZN(IN(LL,8)) - ZC) / ZC + 1.
1053 X1(4) = (XN(IN(LL,4)) - XC) / XC
1054 Y1(4) = ((YN(IN(LL,4)) - YC)/YC)
1055 Z1(4) = (ZN(IN(LL,4)) - ZC) / ZC + 1.
1056 C *****
1057 CALL IGMA(X1(1), Y1(1), Z1(1))
1058 C *****
1059 DO 2130 L1 = 1, 4
1060 C *****
1061 CALL IGDA(X1(L1), Y1(L1), Z1(L1))
1062 C *****
1063 2130 CONTINUE
1064 C *****
1065 CALL IGDA(X1(1), Y1(1), Z1(1))
1066 C *****
1067 2140 CONTINUE
1068 2150 CONTINUE
1069 C *****
1070 CALL IGENDS('PICT')
1071 C *****
1072 CALL IGHUE('PICT', COL(ITR))
1073 2160 CONTINUE
1074 C *****
1075 CALL IGDRON('TERMINAL')
1076 C *****
1077 NPTS1 = NPTS + 1
1078 NELM1 = NELM + 1
1079 2170 WRITE (6,2180)
1080 2180 FORMAT ('To view the results of the 3-D generation, it is', /,
1081 1 'necessary to rotate the picture on the screen. The dir-',
1082 2 /, 'ection of rotation is determined by the "RIGHT HAND RULE
1083 3"', /, 'ie; Point the thumb of the right hand in the positive', /,
1084 4 'direction of the axis(Z into the screen is positive)', /,
1085 5 'The fingers of the right Hand will curl in the positive',
1086 6 /, 'direction of rotation', //, 'Enter the angles in degree
1087 7s', /, 'ANGLEX ')
1088 READ (5,2190) ANGLEX
1089 2190 FORMAT (F7.2)
1090 ANGLEX = (ANGLEX/180) * 3.1416
1091 WRITE (6,2200)
1092 2200 FORMAT ('ANGLEX')
1093 READ (5,2190) ANGLEZ
1094 ANGLEZ = (ANGLEZ/180.) * 3.1416
1095 WRITE (6,2210)
1096 2210 FORMAT ('ANGLEZ')
1097 READ (5,2190) ANGLEZ
1098 ANGLEZ = (ANGLEZ/180.) * 3.1416
1099 C *****
1100 CALL IGTAN('PICT', 'CURRENT', 'ROTX', ANGLEX)

```



```

1101 C *****
1102 C CALL IGTRAN('PICT', 'CURRENT', 'ROTY', ANGLEZ)
1103 C *****
1104 C CALL IGTRAN('PICT', 'CURRENT', 'ROTZ', ANGLEZ)
1105 C *****
1106 C CALL IGDRON('TERMINAL')
1107 C *****
1108 C CALL ASKPLT
1109 C *****
1110 C WRITE (6,2220)
1111 C 2220 FORMAT ('DO YOU WISH TO SEE THIS VIEW IN PERSPECTIVE?')
1112 C READ (5,1610) IANSWR
1113 C IF (IANSWR.NE. YES) GO TO 2230
1114 C *****
1115 C CALL IGTRAN('PICT', 'CURRENT', 'PROJ', 0.0, 0.5)
1116 C *****
1117 C CALL IGDRON('TERMINAL')
1118 C *****
1119 C 2230 CALL ASKPLT
1120 C *****
1121 C WRITE (6,2240)
1122 C 2240 FORMAT ('Do you wish to rotate this figure some more?')
1123 C READ (5,1610) IANSWR
1124 C *****
1125 C CALL IGCTRL('TERMINAL', 'ERASE')
1126 C *****
1127 C IF (IANSWR.NE. YES) GO TO 350
1128 C GO TO 2170
1129 C
1130 C
1131 C SWITCH NO.14
1132 C 2250 CONTINUE
1133 C GO TO 350
1134 C
1135 C 2260 STOP
1136 C END
1137 C
1138 C
1139 C .....
1140 C
1141 C Subroutine SMOOTH
1142 C
1143 C Author
1144 C G.J. Smith(U. of Alberta)
1145 C
1146 C Last Update
1147 C May 1984
1148 C
1149 C Purpose
1150 C To set up the concentration of elements around

```



```

1151 C      a pumping well and smooth out the distribution
1152 C      of elements to the pumping well.
1153 C
1154 C      Usage
1155 C      CALL SMOOTH
1156 C
1157 C      Subroutines Required
1158 C      CNCTQ,CNCTT1,CNCTT2
1159 C
1160 C
1161 C      .....
1162 C      SUBROUTINE SMOOTH
1163 C      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
1164 C      1 ZN(5000)
1165 C      COMMON CX(5000)
1166 C      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
1167 C      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
1168 C      1 ALEAK(100)
1169 C      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
1170 C      1 PPOSZ(10)
1171 C      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
1172 C      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
1173 C      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
1174 C      COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
1175 C      1 XC, IPN
1176 C      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
1177 C      1 DELTZ
1178 C
1179 C      THIS SUBROUTINE SMOOTHES OUT THE RESULTANT MESH USING
1180 C      AN AVERAGING TECHNIQUE
1181 C      10 FORMAT (3I5)
1182 C      K = 1
1183 C      IMM = 1
1184 C      II = 1
1185 C      IFLG = 0
1186 C      IF (ITRNS .LT. 2) IP1 = 1
1187 C      I = NPTS1
1188 C      DO 30 IM = IP1, IP
1189 C      DO 20 IMM = 1, IW2
1190 C      IMM1 = IMM
1191 C      IF (ID .GT. 0) IMM1 = IMM + 1
1192 C      I2 = NDWEL(IM,2,IMM1)
1193 C      IF (I2 .EQ. 0) GO TO 30
1194 C      *****
1195 C      IF (IELEM .GT. 0) CALL CNCTQ(I2, IM, IMM1)
1196 C      *****
1197 C      IF (IELEM .LT. 1) CALL CNCTT1(I2, IM, IMM1)
1198 C      *****
1199 C      20 CONTINUE
1200 C      30 CONTINUE
1201 C      IP1 = IP

```



```

1201 40 CONTINUE
1202   DO 50 LU = 1, NBN
1203     IF (I .EQ. NBDND(LU)) GO TO 140
1204 50 CONTINUE
1205   DO 80 L1 = 1, 2
1206     DO 70 L2 = 1, IP
1207       DO 60 L3 = 1, IW2
1208         IF (I .EQ. NDWEL(L2,L1,L3)) GO TO 140
1209 60 CONTINUE
1210 70 CONTINUE
1211 80 CONTINUE
1212   DO 110 J = 1, NELM
1213     DO 100 L = 1, INC
1214       IF (I .EQ. IN(J,L)) GO TO 90
1215     GO TO 100
1216 90   ISAVE(K,1) = J
1217       K = K + 1
1218     GO TO 110
1219 100 CONTINUE
1220 110 CONTINUE
1221   X = O.
1222   Y = O.
1223   KL = O
1224   KK = K - 1
1225   DO 130 K1 = 1, KK
1226     DO 120 L = 1, INC
1227       N = IN(ISAVE(K1,1),L)
1228       IF (N .EQ. I) GO TO 120
1229       X = XN(N) + X
1230       Y = YN(N) + Y
1231       KL = KL + 1
1232 120 CONTINUE
1233 130 CONTINUE
1234   XN(I) = X / KL
1235   YN(I) = Y / KL
1236   GO TO 140
1237 140 IF (IFLG .EQ. 1) GO TO 150
1238   IF (I .LT. NPTS) GO TO 160
1239   I = NPTS
1240 150 IF (I .LE. NPTS1) GO TO 170
1241   IFLG = 1
1242   K = 1
1243   I = I - 1
1244   GO TO 40
1245 160 I = I + 1
1246   K = 1
1247   GO TO 40
1248 170 CONTINUE
1249   NPTS1 = NPTS + 1
1250   RETURN

```



```

1251      END
1252      C
1253      C
1254      C .....
1255      C
1256      C Subroutine DSTRB
1257      C
1258      C Author
1259      C   G.J. Smith(U. of Alberta)
1260      C
1261      C Last Update
1262      C   May 1984
1263      C
1264      C Purpose
1265      C   To regularly redistribute the the digitized
1266      C   nodes so as to set up the mesh generation
1267      C   process.
1268      C
1269      C
1270      C Usage
1271      C   CALL DSTRB
1272      C
1273      C Subroutines Required
1274      C   None
1275      C
1276      C
1277      C .....
1278      C SUBROUTINE DSTRB
1279      C COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
1280      C 1 ZN(5000)
1281      C COMMON CX(5000)
1282      C COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
1283      C COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
1284      C 1 ALEAK(100)
1285      C COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
1286      C 1 PPOSZ(10)
1287      C COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
1288      C COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
1289      C COMMON ICRNR(10,8), NFACE(6,4), ICNIST(4,2), NODE1(8)
1290      C COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZSTPS, ID,
1291      C 1 YC, XC, IPN
1292      C COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
1293      C 1 DELTZ
1294      C DIMENSION XNEW(5000), YNEW(5000), ZNEW(5000), NFACE1(6,4)
1295      C INTEGER YES /'Y'/, NO /'N'/
1296      C IC1 = ITRNS
1297      C J = NELM + 1
1298      C NZSTPS = 1
1299      C IL = 0
1300      C IC = 0

```



```

1301 IFLG = 0
1302 LC = 0
1303 10 FORMAT ('IN MESH SUBROUTINE')
1304 20 FORMAT (A1)
1305 30 FORMAT (I5)
1306 40 FORMAT ('NPTS=', I5)
1307 50 FORMAT (2I5)
1308 YTOP = AGSENS('TERMINAL','Ysize')
1309 CALL IGTAN('MP*', 'SCALE', YTOP)
1310 60 WRITE (6,70)
1311 70 FORMAT ('The screen will display the digitized figure.', /,
1312           'It is necessary that the user indicate the corners of', /,
1313           'the area that the mesh is to be generated. The user', /,
1314           ', to obtain a well conditioned mesh, must indicate the', /,
1315           ', 4 corners of the area of the tightest mesh first.', /,
1316           'The user will then be prompted to indicate the corners', /
1317           ', of the adjacent portions of differing transmissivities
1318           ', /, 'Touch <RETURN> to continue.')
1319 READ (5,80) KEY
1320 80 FORMAT (I1)
1321 C *****
1322 CALL ICTRL('TERMINAL', 'ERASE')
1323 C *****
1324 CALL IGINIT
1325 C *****
1326 CALL IGBGNS('QUAD')
1327 C *****
1328 CALL IGMA(O.O, O.O)
1329 C *****
1330 X1 = (XN(NPTS1) - XC) / XC
1331 Y1 = ((YN(NPTS1) - YC)/YC)
1332 C *****
1333 CALL IGMA(X1, Y1)
1334 C *****
1335 DO 90 IA = NPTS1, NPTS
1336   X1 = (XN(IA) - XC) / XC
1337   Y1 = ((YN(IA) - YC)/YC)
1338 C *****
1339   CALL IGDA(X1, Y1)
1340 C *****
1341 90 CONTINUE
1342   X1 = (XN(NPTS1) - XC) / XC
1343   Y1 = ((YN(NPTS1) - YC)/YC)
1344 C *****
1345   CALL IGDA(X1, Y1)
1346 C *****
1347   CALL IGDON('TERMINAL')
1348 C *****
1349   IF (ID.LT. 1) GO TO 110
1350   WRITE (6,100)

```



```

1351 100 FORMAT ('Move the cursor with the joystick to indicate the 8', /,
1352 1 'corners of the figure in which the mesh is to be generated
1353 2.', /, 'Enter the lower surface first in the following order: -LOW
1354 3ER', /, 'LEFT-, -LOWER RIGHT-, -UPPER RIGHT-, -UPPER LEFT-. Touch a
1355 4 , /, 'different key to enter each corner and then do the up
1356 5per
1357 6continue.')
1358 GO TO 130
1359
1360 110 WRITE (6,120)
1361 120 FORMAT ('Move the cursor with the joystick to indicate the 4', /,
1362 1 ' , corners of the area in which the mesh is to be generated.
1363 2', /, 'Enter -LOWER LEFT-, -LOWER RIGHT-, -UPPER RIGHT-, -UPPER ',
1364 3 /, 'LEFT-., /, 'Touch a different key to enter each corner
1365 4. Touch <RETURN>., /, ' to continue')
1366
1367 130 NCN = 4
1368 READ (5,30) IANSWR
1369 C *****
1370 CALL ICTRL('TERMINAL', 'ERASE')
1371 C *****
1372 IF (ID .GT. 0) NCN = 8
1373 140 DO 210 IC = 1, NCN
1374 C *****
1375 CALL IGXYIN(XIN(IC), YIN(IC))
1376 C *****
1377 XN(NPTS + IC) = XIN(IC) * XC + XC
1378 YN(NPTS + IC) = (YIN(IC)*YC) + YC
1379 IF (ID .LT. 1) GO TO 180
1380 WRITE (6,160)
1381 160 FORMAT ('Elevation?')
1382 READ (5,170) ZN(NPTS + IC)
1383 170 FORMAT (F10.0)
1384 WRITE (6,190)
1385 190 FORMAT ('IS THIS CORRECT?')
1386 READ (5,20) IANSWR
1387 IF (IANSWR .EQ. NO) GO TO 150
1388 200 FORMAT (3F13.3)
1389 210 CONTINUE
1390 WRITE (6,220)
1391 220 FORMAT ('Did all of the 4(8) corners get input ', /,
1392 1 ' ,corectly? If no, redo all 4(8).')
1393 READ (5,20) IANSWR
1394 IF (IANSWR .EQ. NO) GO TO 140
1395 C *****
1396 CALL IGENDS('QUAD')
1397 C *****
1398 CALL IGDELS('QUAD')
1399 C *****
1400 C Generate NFACE(# of faces) array
1401 IFC = 1
1402 DO 230 M = 1, 4

```



```

1401 NFACE(IFC,M) = NPTS + M
1402 NFACE1(IFC,M) = M
1403 NFACE1(IFC + 1,M) = M + 4
1404
230 CONTINUE
1405 IF (ID .LT. 1) GO TO 250
1406 DO 240 M = 1, 4
1407   NFACE(IFC + 1,M) = NPTS + M + 4
1408
240 CONTINUE
1409 NFACE(3,1) = NPTS + 5
1410 NFACE(3,2) = NPTS + 1
1411 NFACE(3,3) = NPTS + 2
1412 NFACE(3,4) = NPTS + 6
1413 NFACE(4,1) = NPTS + 6
1414 NFACE(4,2) = NPTS + 2
1415 NFACE(4,3) = NPTS + 3
1416 NFACE(4,4) = NPTS + 7
1417 NFACE(5,1) = NPTS + 7
1418 NFACE(5,2) = NPTS + 3
1419 NFACE(5,3) = NPTS + 4
1420 NFACE(5,4) = NPTS + 8
1421 NFACE(6,1) = NPTS + 8
1422 NFACE(6,2) = NPTS + 4
1423 NFACE(6,3) = NPTS + 1
1424 NFACE(6,4) = 1
1425 NFACE1(6,2) = 5
1426 NFACE1(6,3) = 6
1427 NFACE1(6,4) = 2
1428 NFACE1(3,1) = 2
1429 NFACE1(3,2) = 6
1430 NFACE1(3,3) = 7
1431 NFACE1(3,4) = 3
1432 NFACE1(4,1) = 3
1433 NFACE1(4,2) = 7
1434 NFACE1(4,3) = 8
1435 NFACE1(4,4) = 4
1436 NFACE1(5,1) = 4
1437 NFACE1(5,2) = 8
1438 NFACE1(5,3) = 5
1439 NFACE1(5,4) = 1
1440 NFACE(6,4) = NPTS + 5
250 CONTINUE
1441
1442 WRITE (6,260)
1443 260 FORMAT ('Enter the number of elements to be generated in each',/,
1444 'of the X,Y, and Z directions. Remember that common sides must',
1445 '1 /, 'have the same number of elements along their common sides.',
1446 '2 /, 'of X-elements= ?')
1447 READ (5,50) NXSTPS
1448 WRITE (6,280)
1449 280 FORMAT ('# of Y-elements= ?')
1450 READ (5,50) NYSTPS

```



```

1451 IF (ID .LT. 1) GO TO 300
1452 WRITE (6,290)
1453 290 FORMAT ('# of Z-elements= ?')
1454 READ (5,50) NZSTPS
1455 300 WRITE (6,190)
1456 READ (5,20) IANSWR
1457 IF (IANSWR .EQ. NO) GO TO 250
1458 IFCC = 6
1459 IF (ID .LT. 1) IFCC = 1
1460 I = NPTS1
1461 II = NPTS1
1462 IC = 0
1463 DO 1150 IFC = 1, IFCC
1464 IF (ITRNS .LT. 2) GO TO 780
1465 IND = 0
1466 DO 310 M = 1, 4
1467 ICNTST(M,1) = 0
1468 ICNTST(M,2) = 0
1469 310 CONTINUE
1470 IF (ID .GT. 0) GO TO 340
1471 DO 330 M = 1, 4
1472 DO 320 N = 1, NBN
1473 IF (XN(NFACE(IFC,M)) .LT. (XN(NBDND(N)) - 0.5*DELX) .OR. XN(
1474 NFACE(IFC,M)) .GT. (XN(NBDND(N)) + 0.5*DELX))
1475 GO TO 320
1476 IF (YN(NFACE(IFC,M)) .LT. (YN(NBDND(N)) - 0.5*DELY) .OR. YN(
1477 NFACE(IFC,M)) .GT. (YN(NBDND(N)) + 0.5*DELY))
1478 GO TO 320
1479 NFACE(IFC,M) = NBDND(N)
1480 ICRNR(IC1,M) = NBDND(N)
1481 CONTINUE
1482 CONTINUE
1483 GO TO 780
1484 IF (IFC .GE. 2) GO TO 780
1485 DO 440 M = 1, 4
1486 NFN = IFE(IFC)
1487 DO 430 N = 1, NFN
1488 DO 350 MU = 1, 4
1489 MU1 = NFACE1(IFC,MU)
1490 IFN = IN(IFACE(N,IFC),MU1)
1491 IF (XN(NFACE(IFC,M)) .LT. (XN(IFN) - 0.5*DELX) .OR. XN(
1492 NFACE(IFC,M)) .GT. (XN(IFN) + 0.5*DELX)) GO TO 350
1493 IF (YN(NFACE(IFC,M)) .LT. (YN(IFN) - 0.5*DELY) .OR. YN(
1494 NFACE(IFC,M)) .GT. (YN(IFN) + 0.5*DELY)) GO TO 350
1495 IF (ID .LT. 1) GO TO 360
1496 IF (ZN(NFACE(IFC,M)) .LT. (ZN(IFN) - 0.5*DELZ) .OR. ZN(
1497 NFACE(IFC,M)) .GT. (ZN(IFN) + 0.5*DELZ)) GO TO 350
1498 GO TO 360
1499 CONTINUE
1500 GO TO 430

```



```

1501      IND = IND + 1
1502      ICNTST(M,2) = IFACE(N,IFC)
1503      ICNTST(IND,1) = M
1504      NFACE(IFC,M) = IFN
1505      IF (ID .LT. 1) GO TO 420
1506      GO TO (370, 380, 390, 400), M
1507      NFACE(3,2) = IFN
1508      NFACE(6,3) = IFN
1509      GO TO 410
1510      NFACE(3,3) = IFN
1511      NFACE(4,2) = IFN
1512      GO TO 410
1513      NFACE(4,3) = IFN
1514      NFACE(5,2) = IFN
1515      GO TO 410
1516      NFACE(5,3) = IFN
1517      NFACE(6,2) = IFN
1518      CONTINUE
1519      IF (IFC .GT. 2) GO TO 440
1520      IC = IC + 1
1521      ICRNR(IC1,M) = IFN
1522      GO TO 440
1523      CONTINUE
1524      CONTINUE
1525      C Redistribute the nodes along each boundary as a basis for a
1526      C well conditioned grid
1527      IF (ID .LT. 1) GO TO 780
1528      IF (IND .EQ. 4) IFE(IFC) = 0
1529      GO TO (450, 460, 470, 480, 490, 500), IU1
1530      GO TO 680
1531      DO 670 M = 1, 4
1532      IF (ICNTST(M,2) .EQ. 0) GO TO 460
1533      GO TO 670
1534      IU1 = ICNTST(1,1)
1535      GO TO (470, 520, 570, 620), M
1536      GO TO (670, 480, 490, 500), IU1
1537      IU = ICNTST(1,1) + NXSTPS - 1
1538      GO TO 510
1539      IU = ICNTST(1,1) - (NXSTP1 + 1) * (NYSTP1 - 1) - (NXSTPS - 1)
1540      GO TO 510
1541      IU = ICNTST(1,1) - (NXSTP1 + 1) * (NYSTPS - 1)
1542      NFACE(IFC,M) = IN(IU,2)
1543      ICRNR(IC1,M) = IN(IU,2)
1544      NFACE(3,2) = IN(IU,2)
1545      NFACE(6,3) = IN(IU,2)
1546      IC = IC + 1
1547      GO TO 670
1548      GO TO (530, 670, 540, 550), IU1
1549      IU = ICNTST(1,1) + NYSTPS + 1
1550      GO TO 560

```



```

1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600

540  IU = ICNTST(1,1) - (NXSTP1 + 1) * (NYSTPS - 1)
      GO TO 560
550  IU = ICNTST(1,1) - ((NXSTP1 + 1)*(NYSTP1 - 1) - 1)
560  NFACE(IFC,M) = IN(IU,3)
      ICRNR(IC1,M) = IN(IU,3)
      NFACE(3,3) = IN(IU,3)
      NFACE(4,2) = IN(IU,3)
      IC = IC + 1
      GO TO 670
570  GO TO (580, 590, 670, 600), IU1
580  IU = ICNTST(1,1) - ((NXSTP1 + 1)*(NYSTPS - 2) + NXSTPS - 1)
      GO TO 610
590  IU = ICNTST(1,1) - (NXSTP1 + 1) * (NYSTP1 + 1)
      GO TO 610
600  IU = ICNTST(1,1) - (NXSTPS - 1)
610  NFACE(IFC,M) = IN(IU,4)
      ICRNR(IC1,M) = IN(IU,4)
      NFACE(4,3) = IN(IU,4)
      NFACE(5,2) = IN(IU,4)
      GO TO 670
620  GO TO (630, 640, 650, 670), IU1
630  IU = ICNTST(1,1) - (NXSTP1 + 1) * (NYSTPS - 1)
      GO TO 660
640  IU = ICNTST(1,1) - ((NXSTP1 + 1 - NXSTPS) + (NXSTP1 + 1))*
      1 NYSTPS - 1)
      GO TO 660
650  IU = ICNTST(1,1) + NXSTPS
660  NFACE(IFC,M) = IN(IU,1)
      ICRNR(IC1,M) = IN(IU,1)
      NFACE(5,3) = IN(IU,1)
      NFACE(6,2) = IN(IU,1)
      IC = IC + 1
      CONTINUE
670  CONTINUE
680  CONTINUE
      GO TO 770
690  DO 700 M = 1, 4
      IF (ICNTST(M,2) .EQ. 0) GO TO 710
700  CONTINUE
      GO TO 770
710  GO TO (720, 730, 740, 750), M
720  IU = ICNTST(2,2) - NXSTPS - 1
      NFACE(IFC,M) = IN(IU,2)
      ICRNR(IC1,M) = IN(IU,2)
      NFACE(3,2) = IN(IU,2)
      NFACE(6,3) = IN(IU,2)
      IC = IC + 1
      GO TO 760
730  IU = ICNTST(1,2) + NXSTPS - 1
      NFACE(IFC,M) = IN(IU,3)
      ICRNR(IC1,M) = IN(IU,3)

```



```

1601 NFACE(3,3) = IN(IU,3)
1602 NFACE(4,2) = IN(IU,3)
1603 IC = IC + 1
1604 GO TO 760
1605 IU = ICNTST(4,2) + NXSTPS - 1
1606 NFACE(IFC,M) = IN(IU,4)
1607 ICRNR(IC1,M) = IN(IU,4)
1608 NFACE(4,3) = IN(IU,4)
1609 NFACE(5,2) = IN(IU,4)
1610 IC = IC + 1
1611 GO TO 760
1612 IU = ICNTST(3,2) - NXSTPS - 1
1613 NFACE(IFC,M) = IN(IU,1)
1614 ICRNR(IC1,M) = IN(IU,1)
1615 NFACE(5,3) = IN(IU,1)
1616 NFACE(6,2) = IN(IU,1)
1617 IC = IC + 1
1618 CONTINUE
1619 IFE(IFC) = 0
1620 GO TO 1150
1621 CONTINUE
1622 DZ = 0.0
1623 DO 1140 L = 1, 4
1624 L1 = L
1625 L2 = L + 1
1626 IF (IFC.GT. 2.AND. L.GT. 1) GO TO 1150
1627 IF (L.EQ. 4) GO TO 800
1628 GO TO 810
1629 L1 = 4
1630 L2 = 1
1631 IF (IFC.GT. 2) GO TO 820
1632 IF (ID.LT. 1.AND. NDX(NFACE(IFC,L1)).NE. 0) IC = IC + 1
1633 IF (L1.LT. 3.AND. NDX(NFACE(IFC,L1)).NE. 0) GO TO 1140
1634 IF (ID.LT. 1.AND. NDY(NFACE(IFC,L1)).NE. 0) IC = IC + 1
1635 IF (NDY(NFACE(IFC,L1)).NE. 0) GO TO 1140
1636 IF (ID.LT. 1.AND. NDZ(NFACE(IFC,L2)).NE. 0) IC = IC + 1
1637 IF (NDZ(NFACE(IFC,L2)).NE. 0) GO TO 1140
1638 DX = XN(NFACE(IFC,L2)) - XN(NFACE(IFC,L1))
1639 DY = YN(NFACE(IFC,L2)) - YN(NFACE(IFC,L1))
1640 IF (ID.LT. 1) GO TO 840
1641 DZ = ZN(NFACE(IFC,L2)) - ZN(NFACE(IFC,L1))
1642 IF (IFC.GT. 2) DZ = -DZ
1643 IF (IFC.GT. 2) GO TO 870
1644 IF (ABS(DX).GT. ABS(DY).AND. ABS(DX).GT. ABS(DZ))
1645 1 GO TO 850
1646 IF (ABS(DY).GT. ABS(DX).AND. ABS(DY).GT. ABS(DZ))
1647 1 GO TO 860
1648 IF (ABS(DZ).GT. ABS(DX).AND. ABS(DZ).GT. ABS(DY))
1649 1 GO TO 870
1650 IF (ABS(DX).GT. ABS(DY)) GO TO 850

```



```

1651 GO TO 860
1652 XPT = XN(NFACE(IFC,L1))
1653 DELX = DX / NXSTPS
1654 NXSTP1 = NXSTPS - 1
1655 YPT = YN(NFACE(IFC,L1))
1656 IF (ID .LT. 1) GO TO 880
1657 DELZ = DZ / NXSTPS
1658 ZPT = ZN(NFACE(IFC,L1))
1659 GO TO 880
1660 YPT = YN(NFACE(IFC,L1))
1661 DELY = DY / NYSTPS
1662 NYSTP1 = NYSTPS - 1
1663 XPT = XN(NFACE(IFC,L1))
1664 IF (ID .LT. 1) GO TO 990
1665 ZPT = ZN(NFACE(IFC,L1))
1666 DELZ = DZ / NYSTPS
1667 GO TO 990
1668 ZPT = ZN(NFACE(IFC,L2))
1669 P1 = NFACE(IFC,L1)
1670 P2 = NFACE(IFC,L2)
1671 DELZ = DZ / NZSTPS
1672 NZSTP1 = NZSTPS - 1
1673 XPT = XN(NFACE(IFC,L2))
1674 YPT = YN(NFACE(IFC,L2))
1675 LC = LC + 1
1676 NDZ(ICRNR(IC1,LC)) = I
1677 GO TO 1100
1678 XNEW(I) = XPT
1679 ZNEW(I) = ZPT
1680 YNEW(I) = YPT
1681 IF (ID .LT. 1 .AND. DELX .LT. 0.0 .AND. IC .GT. 2)
1682 GO TO 890
1683 IC = IC + 1
1684 ICRNR(IC1,IC) = I
1685 NBN = NBN + 1
1686 NBDND(NBN) = I
1687 I = I + 1
1688 DO 980 INX = 1, NXSTP1
1689 XPT = XPT + DELX
1690 I11 = I1 + 1
1691 I12 = I1 + 2
1692 D1 = XN(I1)
1693 D2 = XN(I11)
1694 D3 = XN(I12)
1695 D4 = YN(I1)
1696 D5 = YN(I11)
1697 D6 = YN(I12)
1698 IF (XN(I11) .LT. (XN(I12) - 0.01*SCL) .OR. XN(I11) .GT. (XN(
1699 I12) + 0.01*SCL)) GO TO 910
1700 IF (YN(I11) .LT. (YN(I12) - 0.01*SCL) .OR. YN(I11) .GT. (YN(

```



```

1701      1      II2) + 0.01*SCL)) GO TO 910
1702      II1 = II1 + 1
1703      II2 = II2 + 1
1704      D2 = XN(II1)
1705      D3 = XN(II2)
1706      D5 = YN(II1)
1707      D6 = YN(II2)
1708      IF (XN(II1) .LT. (XN(II1) - 0.01*SCL) .OR. XN(II1) .GT. (XN(
1709      1      II1) + 0.01*SCL)) GO TO 920
1710      IF (YN(II1) .LT. (YN(II1) - 0.01*SCL) .OR. YN(II1) .GT. (YN(
1711      1      II1) + 0.01*SCL)) GO TO 920
1712      GO TO 930
1713      IF (DELX .LT. 0.0) GO TO 940
1714      IF (XPT .GT. D2) GO TO 930
1715      GO TO 950
1716      II = II + 1
1717      II1 = II + 1
1718      II2 = II + 2
1719      GO TO 900
1720      IF (XPT .LT. D2) GO TO 930
1721      IF (INX .EQ. 1 .AND. XPT .LT. D3) GO TO 930
1722      CONTINUE
1723      C Lagrange interpolation routine
1724      A = (XPT - D2) * (XPT - D3)
1725      B = (D1 - D2) * (D1 - D3)
1726      P1 = (A/B) * D4
1727      C = (XPT - D1) * (XPT - D3)
1728      D = (D2 - D1) * (D2 - D3)
1729      P2 = (C/D) * D5
1730      E = (XPT - D1) * (XPT - D2)
1731      F = (D3 - D1) * (D3 - D2)
1732      P3 = (E/F) * D6
1733      YNEW(I) = P1 + P2 + P3
1734      XNEW(I) = XPT
1735      CX(I) = CX(ITRNS)
1736      IF (ID .LT. 1) GO TO 960
1737      ZNEW(I) = ZNEW(I - 1) + DELZ
1738      NBN = NBN + 1
1739      NBDND(NBN) = I
1740      IF (DELX .LT. 0.00) GO TO 970
1741      NDX(I - 1) = I
1742      IF (INX .EQ. 1) NDX(ICRNR(IC1,IC)) = I
1743      IF (IFC .LT. 3 .AND. L .EQ. 4) NDY(I) = I - 1
1744      IF (IFC .LT. 3 .AND. L .EQ. 4 .AND. INX .EQ. 1)
1745      1      NDY(I) = ICRNR(IC1,IC)
1746      IF (IFC .GT. 2 .AND. L .EQ. 2) NDZ(I - 1) = I
1747      IF (IFC .GT. 2 .AND. L .EQ. 2 .AND. INX .EQ. 1)
1748      1      NDZ(ICRNR(IC1,IC)) = I
1749      I = I + 1
1750      GO TO 980

```



```

1751 NDX(I) = I - 1
1752 IF (INX .EQ. 1) NDX(ICRNR(IC1,IC)) = I - 1
1753 IF (IFC .LT. 3 .AND. L .EQ. 2) NDY(I - 1) = I
1754 IF (IFC .LT. 3 .AND. L .EQ. 2 .AND. INX .EQ. 1)
1755 NDY(ICRNR(IC1,IC)) = I
1756 IF (IFC .GT. 2 .AND. L .EQ. 4) NDZ(I) = I - 1
1757 IF (IFC .GT. 2 .AND. L .EQ. 4 .AND. INX .EQ. 1)
1758 NDZ(I) = ICRNR(IC1,IC)
1759 I = I + 1
1760 CONTINUE
1761 GO TO 1140
1762 YNEW(I) = YPT
1763 XNEW(I) = XPT
1764 ZNEW(I) = ZPT
1765 IC = IC + 1
1766 ICRNR(IC1,IC) = I
1767 NBN = NBN + 1
1768 NBDND(NBN) = I
1769 I = I + 1
1770 DO 1090 INY = 1, NYSTP1
1771 YPT = YPT + DELY
1772 II1 = II + 1
1773 II2 = II + 2
1774 D1 = YN(II)
1775 D2 = YN(II1)
1776 D3 = YN(II2)
1777 D4 = XN(II)
1778 D5 = XN(II1)
1779 D6 = XN(II2)
1780 IF (XN(II1) .LT. (XN(II2) - 0.01*SCL) .OR. XN(II1) .GT. (XN(
1781 II2) + 0.01*SCL)) GO TO 1010
1782 IF (YN(II1) .LT. (YN(II2) - 0.01*SCL) .OR. YN(II1) .GT. (YN(
1783 II2) + 0.01*SCL)) GO TO 1010
1784 II1 = II1 + 1
1785 II2 = II2 + 1
1786 D2 = YN(II1)
1787 D3 = YN(II2)
1788 D5 = XN(II1)
1789 D6 = XN(II2)
1790 IF (XN(II) .LT. (XN(II1) - 0.01*SCL) .OR. XN(II) .GT. (XN(
1791 II1) + 0.01*SCL)) GO TO 1020
1792 IF (YN(II) .LT. (YN(II1) - 0.01*SCL) .OR. YN(II) .GT. (YN(
1793 II1) + 0.01*SCL)) GO TO 1020
1794 GO TO 1030
1795 IF (DELY .LT. 0.0) GO TO 1040
1796 IF (YPT .GT. YN(II + 1)) GO TO 1030
1797 GO TO 1050
1798 II = II + 1
1799 GO TO 1000
1800 IF (YPT .LT. YN(II + 1)) GO TO 1030
970
1
1
1
980
990
1000
1
1
1010
1
1
1020
1030
1040

```



```

1801      1050      CONTINUE
1802      IF (II .LT. NPTS - 2) GO TO 1060
1803      D1 = YN(NPTS - 2)
1804      D2 = YN(NPTS - 1)
1805      D3 = YN(NPTS)
1806      D4 = XN(NPTS - 2)
1807      D5 = XN(NPTS - 1)
1808      D6 = XN(NPTS)
1809      C Lagrange interpolation routine
1810      1060      A = (YPT - D2) * (YPT - D3)
1811      B = (D1 - D2) * (D1 - D3)
1812      P1 = (A/B) * D4
1813      C = (YPT - D1) * (YPT - D3)
1814      D = (D2 - D1) * (D2 - D3)
1815      P2 = (C/D) * D5
1816      E = (YPT - D1) * (YPT - D2)
1817      F = (D3 - D1) * (D3 - D2)
1818      P3 = (E/F) * D6
1819      XNEW(I) = P1 + P2 + P3
1820      YNEW(I) = YPT
1821      IF (ID .LT. 1) GO TO 1070
1822      ZNEW(I) = ZNEW(I - 1) + DELZ
1823      CX(I) = CX(ITRNS)
1824      NBN = NBN + 1
1825      NBDND(NBN) = I
1826      IF (DELY .LT. 0.00) GO TO 1080
1827      NDY(I - 1) = I
1828      IF (INY .EQ. 1) NDY(ICRNR(IC1, IC)) = I
1829      IF (IFC .LT. 3 .AND. L .EQ. 3) NDX(I) = I - 1
1830      IF (IFC .LT. 3 .AND. L .EQ. 3 .AND. INY .EQ. 1)
1831      1      NDY(I) = ICRNR(IC1, IC)
1832      IF (IFC .GT. 2 .AND. L .EQ. 1) NDZ(I - 1) = I
1833      IF (IFC .GT. 2 .AND. L .EQ. 1 .AND. INY .EQ. 1)
1834      1      NDZ(ICRNR(IC1, IC)) = I
1835      I = I + 1
1836      GO TO 1090
1837      1080      NDY(I) = I - 1
1838      IF (INY .EQ. 1) NDY(I) = ICRNR(IC1, IC)
1839      IF (IFC .LT. 3 .AND. L .EQ. 1) NDX(I - 1) = I
1840      IF (IFC .LT. 3 .AND. L .EQ. 1 .AND. INY .EQ. 1)
1841      1      NDX(ICRNR(IC1, IC)) = I
1842      IF (IFC .GT. 2 .AND. L .EQ. 3) NDZ(I) = I - 1
1843      IF (IFC .GT. 2 .AND. L .EQ. 3 .AND. INY .EQ. 1)
1844      1      NDZ(I) = ICRNR(IC1, IC)
1845      I = I + 1
1846      CONTINUE
1847      GO TO 1140
1848      IF (NZSTPS .LT. 2) GO TO 1130
1849      DO 1120 INZ = 1, NZSTP1
1850      XNEW(I) = XPT + ((XN(P1) - XN(P2))/NZSTPS) * INZ

```



```

1851 YNEW(I) = YPT + ((YN(P1) - YN(P2))/NZSTPS) * INZ
1852 ZNEW(I) = ZPT + DELZ * (INZ)
1853 CX(I) = CX(ITRNS)
1854 NBN = NBN + 1
1855 NBDND(NBN) = I
1856 IF (DELZ .LT. 0.00) GO TO 1110
1857 IF (INZ .GT. 1) NDZ(I - 1) = I
1858 I = I + 1
1859 GO TO 1120
1860 NDZ(I) = I - 1
1861 IF (INZ .EQ. 1) NDZ(I) = ICRNR(IC1,LC)
1862 I = I + 1
1863 CONTINUE
1864 NDZ(I - 1) = ICRNR(IC1,LC + 4)
1865 GO TO 1140
1866 NDZ(ICRNR(IC1,LC)) = ICRNR(IC1,LC + 4)
1867 CONTINUE
1868 CONTINUE
1869 FORMAT (I5, 3F13.3)
1870 I = I - 1
1871 DO 1170 I1 = NPTS1, I
1872   XN(I1) = XNEW(I1)
1873   YN(I1) = YNEW(I1)
1874   IF (ID .LT. 1) GO TO 1170
1875   ZN(I1) = ZNEW(I1)
1876 CONTINUE
1877 NPTS = I
1878 NDX(ICRNR(IC1,4) - (NXSTPS - 1)) = ICRNR(IC1,3)
1879 NDX(ICRNR(IC1,4)) = ICRNR(IC1,4) - 1
1880 IF (IC1 .LT. 2) NDY(ICRNR(IC1,1)) = ICRNR(IC1,4) + NYSTPS - 1
1881 IF (ID .LT. 1) NDY(ICRNR(IC1,1)) = ICRNR(IC1,4) + NYSTPS - 1
1882 NDX(ICRNR(IC1,1) + NXSTPS - 1) = ICRNR(IC1,2)
1883 NDY(ICRNR(IC1,2) + NYSTPS - 1) = ICRNR(IC1,3)
1884 IF (ID .LT. 1) GO TO 1180
1885 NDX(ICRNR(IC1,6) - 1) = ICRNR(IC1,6)
1886 NDX(ICRNR(IC1,8)) = ICRNR(IC1,8) - 1
1887 NDY(ICRNR(IC1,5)) = ICRNR(IC1,8) + NYSTP1
1888 NDY(ICRNR(IC1,7) - 1) = ICRNR(IC1,7)
1889 CONTINUE
1890 C *****
1891   CALL NODE(IC1)
1892 C *****
1893   RETURN
1894   END
1895 C
1896 C
1897 C .....
1898 C Subroutine NODE
1899 C
1900 C

```



```

1901 C Author
1902 C G.J. Smith(U. of Alberta)
1903 C
1904 C Last Update
1905 C
1906 C May 1984
1907 C
1908 C Purpose
1909 C To generate the nodes within the digitized
1910 C figure to form the mesh.
1911 C
1912 C Usage
1913 C CALL NODE(IC1)
1914 C
1915 C Subroutines Required
1916 C ELMNT
1917 C
1918 C
1919 C
1920 C .....
1921 C SUBROUTINE NODE(IC1)
1922 C COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
1923 C 1 ZN(5000)
1924 C COMMON CX(5000)
1925 C COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
1926 C COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
1927 C 1 ALEAK(100)
1928 C COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
1929 C 1 PPOSZ(10)
1930 C COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
1931 C COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
1932 C COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
1933 C COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZSTPS, ID,
1934 C 1 YC, XC, IPN
1935 C COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
1936 C 1 DELTZ
1937 C NDD = ICRNR(IC1,1)
1938 C I = 1
1939 C IFLG = 0
1940 C J = NELM + 1
1941 C NXSTP1 = NXSTPS - 1
1942 C NYSTP1 = NYSTPS - 1
1943 C IF (ID .GT. 0) NZSTP1 = NZSTPS - 1
1944 C IL = 0
1945 C NDDO = ICRNR(IC1,2)
1946 C 10 IL = IL + 1
1947 C ND1 = NDD
1948 C NDD2 = NDY(NDDO)
1949 C IF (ID .GT. 0 .AND. IL .LT. NZSTPS) NDD = NDZ(ND1)
1950 C ICOL = 1
1951 C IROW = 1

```



```

1951 DO 350 IROW = 1, NYSTP1
1952   NODE1(6) = ND1
1953   NODE1(5) = NDY(ND1)
1954   NODE1(7) = NDX(ND1)
1955   IF (ID .GT. 0) NODE1(2) = NDZ(ND1)
1956   NDD1 = NDY(ND1)
1957   ND1 = NODE1(5)
1958   DELTX = XN(NDD2) - XN(NDD1)
1959   DELTX = DELTX / NXSTPS
1960   IF (AMOD(FLOAT(NXSTPS),2.) .GT. 0.00) GO TO 30
1961   IF (IFLG .EQ. 1) GO TO 20
1962   IF (IFLG .EQ. 0) IFLG = 1
1963   GO TO 30
1964   IFLG = 0
1965   CONTINUE
1966   DO 340 ICOL = 1, NXSTP1
1967     IF (ICOL .EQ. 1 .AND. IROW .EQ. 1 .AND. ID .GT. 0)
1968       1 NDD = NODE1(2)
1969     IF (ID .GT. 0) DELTZ = ZN(ICRNR(IC1,5) + ICOL) - ZN(ICRNR(IC1,
1970       1) + ICOL)
1971     IF (ID .GT. 0) DELTZ = DELTZ / NZSTPS
1972     IF (ICOL .EQ. 1) NDD1 = NDY(ND1)
1973     IF (ICOL .EQ. 1) NDD3 = NODE1(5)
1974     DELTY = (YN(ICRNR(IC1,4) - ICOL) - YN(ICOL + ICRNR(IC1,1)))
1975     DELTY = DELTY / NYSTPS
1976     IF (ID .LT. 1) GO TO 40
1977     ZL1 = (ZN(ICRNR(IC1,1)) - ZN(ICRNR(IC1,2))) * (NXSTPS - ICOL)
1978     / NXSTPS + ZN(ICRNR(IC1,2))
1979     ZL2 = (ZN(ICRNR(IC1,4)) - ZN(ICRNR(IC1,3))) * (NXSTPS - ICOL)
1980     / NXSTPS + ZN(ICRNR(IC1,3))
1981     ZL = (ZL1 - ZL2) * (NYSTPS - IROW) / NYSTPS + ZL2
1982     IF ((ID .GT. 0 .AND. IROW .GT. 1) .OR. (ID .GT. 0 .AND. IL .
1983     EQ. NZSTPS)) NODE1(3) = NDX(NODE1(2))
1984     NODE1(7) = NDX(NODE1(6))
1985     IF (NDY(NODE1(7)) .NE. 0) GO TO 50
1986     IF (IL .GT. 1) GO TO 50
1987     GO TO 60
1988     NODE1(8) = NDY(NODE1(7))
1989     GO TO 100
1990     CONTINUE
1991     XN(NPTS + I) = XN(NODE1(5)) + DELTX
1992     YN(NPTS + I) = YN(NODE1(7)) + DELTY
1993     NODE1(8) = NPTS + I
1994     FORMAT (I5, 2F10.3)
1995     IF (ID .GT. 0) ZN(NODE1(8)) = ZL
1996     IF (IL .EQ. 1 .AND. ID .GT. 0) GO TO 80
1997     GO TO 90
1998     NBN = NBN + 1
1999     NBDND(NBN) = NPTS + I
2000     IF (IC1 .GT. 1) GO TO 90

```



```

2001 IFE(2) = IFE(2) + 1
2002 IFACE(IFE(2),2) = J
2003 CONTINUE
2004 IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
2005 NDY(NODE1(5)) = NPTS + I
2006 NDY(NODE1(7)) = NPTS + I
2007 I = I + 1
2008 IF (ID .LT. 1) GO TO 190
2009 IF (IL .EQ. NZSTPS) GO TO 120
2010 IF (NDZ(NODE1(5)) .NE. 0) GO TO 130
2011 IF (ICOL .GT. 1) GO TO 130
2012 XN(NPTS + I) = XN(NODE1(5))
2013 YN(NPTS + I) = YN(NODE1(5))
2014 ZN(NPTS + I) = ZN(NODE1(5)) + DELTZ
2015 IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
2016 NDY(NODE1(2)) = NPTS + I
2017 NDZ(NODE1(5)) = NPTS + I
2018 NODE1(1) = NPTS + I
2019 NBN = NBN + 1
2020 NBDND(NBN) = NPTS + I
2021 IFE(6) = IFE(6) + 1
2022 IFACE(IFE(6),6) = J
2023 CONTINUE
2024 I = I + 1
2025 GO TO 130
2026 NODE1(1) = NDY(NODE1(2))
2027 NDZ(NODE1(5)) = NODE1(1)
2028 NDY(NODE1(2)) = NODE1(1)
2029 IFE(6) = IFE(6) + 1
2030 IFACE(IFE(6),6) = J
2031 IF (IROW .GT. 1 .OR. IL .EQ. NZSTPS) GO TO 140
2032 XN(NPTS + I) = XN(NDX(NODE1(6)))
2033 YN(NPTS + I) = YN(NDX(NODE1(6)))
2034 ZN(NPTS + I) = ZN(NDX(NODE1(6))) + DELTZ
2035 IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
2036 NDZ(NODE1(7)) = NPTS + I
2037 NDY(NODE1(2)) = NPTS + I
2038 IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
2039 NODE1(3) = NPTS + I
2040 NBN = NBN + 1
2041 NBDND(NBN) = NPTS + I
2042 I = I + 1
2043 NODE1(3) = NDY(NODE1(2))
2044 IF (IROW .GT. 1) GO TO 150
2045 IFE(3) = IFE(3) + 1
2046 IFACE(IFE(3),3) = J
2047 IF (NDY(NODE1(3)) .EQ. 0) GO TO 160
2048 NODE1(4) = NDY(NODE1(3))
2049 XN(NPTS + I) = XN(NODE1(8))
2050 YN(NPTS + I) = YN(NODE1(8))

```



```

2051 ZN(NPTS + I) = ZN(NODE1(8)) + DELTZ
2052 NDZ(NODE1(8)) = NPTS + I
2053 NDY(NODE1(3)) = NPTS + I
2054 NDX(NODE1(1)) = NPTS + I
2055 NODE1(4) = NPTS + I
2056 IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
2057 IF (IL .EQ. NZSTPS .AND. ID .GT. 0) GO TO 170
2058 GO TO 180
2059 NBN = NBN + 1
2060 NBDND(NBN) = NPTS + I
2061 IFE(1) = IFE(1) + 1
2062 IFACE(IFE(1),1) = J
2063 CONTINUE
2064 I = I + 1
2065 CONTINUE
2066 IF (IO .LT. 1) GO TO 250
2067 C Look for observation wells
2068 DO 240 IOW = 1, IO
2069   IF (XN(NODE1(8)) .LE. (OBSX(IOW) - DELTX*0.5) .OR. XN(NODE1(
2070     8)) .GT. (OBSX(IOW) + DELTX*0.5)) GO TO 240
2071   IF (YN(NODE1(8)) .LE. (OBSY(IOW) - DELTY*0.5) .OR. YN(NODE1(
2072     8)) .GT. (OBSY(IOW) + DELTY*0.5)) GO TO 240
2073   IF (ID .LT. 1) GO TO 230
2074   DO 200 IWN = 1, IW1
2075     IF (NODE1(4) .EQ. NDWEL(IOW,1,IWN)) GO TO 240
2076     CONTINUE
2077     IW1 = IW1 + 1
2078     XN(NODE1(4)) = OBSX(IOW)
2079     YN(NODE1(4)) = OBSY(IOW)
2080     NDWEL(IOW,1,IW1) = NODE1(4)
2081     DO 220 IWN = 1, IW1
2082       IF (NODE1(8) .EQ. NDWEL(IOW,1,IWN)) GO TO 240
2083       CONTINUE
2084       IW1 = IW1 + 1
2085       XN(NODE1(8)) = OBSX(IOW)
2086       YN(NODE1(8)) = OBSY(IOW)
2087       NDWEL(IOW,1,IW1) = NODE1(8)
2088     CONTINUE
2089     CONTINUE
2090   IF (IP .LT. 1) GO TO 310
2091   C Look for pumping wells
2092   DO 300 IPW = 1, IP
2093     IF (XN(NODE1(8)) .LE. (PPOSX(IPW) - DELTX*0.5) .OR. XN(
2094       NODE1(8)) .GT. (PPOSX(IPW) + DELTX*0.5)) GO TO 300
2095     IF (YN(NODE1(8)) .LE. (PPOSY(IPW) - DELTY*0.5) .OR. YN(
2096       NODE1(8)) .GT. (PPOSY(IPW) + DELTY*0.5)) GO TO 300
2097     IF (ID .LT. 1) GO TO 270
2098     IF (IW2 .LT. 1) GO TO 270
2099     DO 260 IWN = 1, IW2
2100       IF (NODE1(8) .EQ. NDWEL(IPW,2,IWN)) GO TO 280

```



```

2101 CONTINUE
2102 IW2 = IW2 + 1
2103 XN(NODE1(8)) = PPOSX(IPW)
2104 YN(NODE1(8)) = PPOSY(IPW)
2105 NDWEL(IPW,2,IW2) = NODE1(8)
2106 IPPN(IPW) = IW2
2107 ITYPE(NODE1(8)) = 2
2108 IF (ID .LT. 1) GO TO 300
2109 DO 290 IWN = 1, IW2
2110 IF (NODE1(4) .EQ. NDWEL(IPW,2,IWN)) GO TO 300
2111 CONTINUE
2112 XN(NODE1(4)) = PPOSX(IPW)
2113 YN(NODE1(4)) = PPOSY(IPW)
2114 NDWEL(IPW,2,IW2) = NODE1(4)
2115 ITYPE(NODE1(4)) = 2
2116 IPPN(IPW) = IW2
2117 IW2 = IW2 + 1
2118 CONTINUE
2119 CONTINUE
2120 FORMAT (9I5)
2121 FORMAT (4I5)
2122 NPTSI = NPTS + I - 1
2123 C *****
2124 CALL ELMNT(I, J, NPTSI, IROW, ICOL, ND1, NDD2, NDD3, IL, IC1,
2125 1 IFLG)
2126 C *****
2127 CONTINUE
2128 ICOL = ICOL + 1
2129 C *****
2130 CALL ELMNT(I, J, NPTSI, IROW, ICOL, ND1, NDD2, NDD3, IL, IC1,
2131 1 IFLG)
2132 C *****
2133 CONTINUE
2134 NODE1(5) = ICRNR(IC1,4)
2135 NODE1(6) = NODE1(5) + 1
2136 DO 460 IXL = 1, NXSTP1
2137 IF (ID .GT. 0 .OR. IELEM .EQ. 1) GO TO 380
2138 IF (MOD(NXSTPS,2) .GT. 0) GO TO 360
2139 IF (IFLG .EQ. 0) IFLG = 1
2140 IF (IFLG .EQ. 1) IFLG = 0
2141 IF (IFLG .EQ. 1) GO TO 370
2142 IN(J,1) = NODE1(5)
2143 IN(J,2) = NODE1(6)
2144 IN(J,3) = NDX(NODE1(5))
2145 J = J + 1
2146 IN(J,1) = NDX(NODE1(5))
2147 IN(J,2) = NODE1(6)
2148 IN(J,3) = NDX(NODE1(6))
2149 NODE1(5) = IN(J - 1,3)
2150 NODE1(6) = IN(J,3)

```



```

2151 J = J + 1
2152 IFLG = 1
2153 GO TO 460
2154 IN(J,1) = NODE1(5)
2155 IN(J,2) = NODE1(6)
2156 IN(J,3) = NDX(NODE1(6))
2157 J = J + 1
2158 IN(J,1) = NODE1(5)
2159 IN(J,2) = NDX(NODE1(6))
2160 IN(J,3) = NDX(NODE1(5))
2161 NODE1(5) = IN(J,3)
2162 NODE1(6) = IN(J,2)
2163 J = J + 1
2164 IFLG = 0
2165 GO TO 460
2166 NODE1(6) = ND1
2167 NODE1(5) = NDY(ND1)
2168 NODE1(7) = NDX(NODE1(6))
2169 NODE1(8) = NDX(NODE1(5))
2170 NDY(NODE1(7)) = NODE1(8)
2171 IF (ID .LT. 1) GO TO 410
2172 NODE1(1) = NDZ(NODE1(5))
2173 NODE1(2) = NDZ(NODE1(6))
2174 NODE1(3) = NDZ(NODE1(7))
2175 NDX(NODE1(2)) = NODE1(3)
2176 NDY(NODE1(2)) = NODE1(1)
2177 IF (IL .EQ. NZSTPS) GO TO 390
2178 GO TO 400
2179 NODE1(4) = NDX(NODE1(1))
2180 NDY(NODE1(3)) = NODE1(4)
2181 NDZ(NODE1(8)) = NODE1(4)
2182 IFE(1) = IFE(1) + 1
2183 IFACE(IFE(1),1) = J
2184 GO TO 410
2185 CONTINUE
2186 XN(NPTS + I) = XN(NODE1(8))
2187 YN(NPTS + I) = YN(NODE1(8))
2188 ZN(NPTS + I) = ZN(NODE1(8)) + DELTZ
2189 IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
2190 NODE1(4) = NPTS + I
2191 NBN = NBN + 1
2192 NBDND(NBN) = NPTS + I
2193 NDX(NODE1(1)) = NODE1(4)
2194 NDY(NODE1(3)) = NODE1(4)
2195 NDZ(NODE1(8)) = NODE1(4)
2196 I = I + 1
2197 N1 = 5
2198 N2 = 8
2199 IFE(5) = IFE(5) + 1
2200 IFACE(IFE(5),5) = J
370
380
390
400
410

```



```

2201 IF (IL .GT. 1 .OR. IC1 .GT. 1) GO TO 420
2202 IFE(2) = IFE(2) + 1
2203 IFACE(IFE(2),2) = J
2204 IF (IXL .GT. 1) GO TO 430
2205 IFE(6) = IFE(6) + 1
2206 IFACE(IFE(6),6) = J
2207 CONTINUE
2208 L = 0
2209 IF (ID .GT. 0) N1 = 1
2210 DO 440 N = N1, N2
2211   L = L + 1
2212   IN(J,L) = NODE1(N)
2213 CONTINUE
2214 IF (ID .LT. 1) GO TO 450
2215 IFE(5) = IFE(5) + 1
2216 IFACE(IFE(5),5) = J
2217 CONTINUE
2218 J = J + 1
2219 ND1 = NODE1(7)
2220 CONTINUE
2221 IF (IELEM .EQ. 1) GO TO 480
2222 IF (IFLG .EQ. 1) GO TO 470
2223 IN(J,1) = NODE1(5)
2224 IN(J,2) = NODE1(6)
2225 IN(J,3) = ICRNR(IC1,3)
2226 J = J + 1
2227 IN(J,1) = ICRNR(IC1,3)
2228 IN(J,2) = NODE1(6)
2229 IN(J,3) = NDX(NODE1(6))
2230 J = J + 1
2231 GO TO 530
2232 IN(J,1) = NODE1(5)
2233 IN(J,2) = NODE1(6)
2234 IN(J,3) = NDX(NODE1(6))
2235 J = J + 1
2236 IN(J,1) = NODE1(5)
2237 IN(J,2) = NDX(NODE1(6))
2238 IN(J,3) = ICRNR(IC1,3)
2239 J = J + 1
2240 GO TO 530
2241 NODE1(6) = ND1
2242 NODE1(5) = NDY(ND1)
2243 NODE1(7) = NDX(ND1)
2244 NODE1(8) = NDX(NODE1(5))
2245 NODE1(2) = NDZ(ND1)
2246 NODE1(1) = NDZ(NODE1(5))
2247 NODE1(3) = NDZ(NODE1(7))
2248 NODE1(4) = NDZ(NODE1(8))
2249 NDX(NODE1(1)) = NODE1(4)
2250 N1 = 5

```



```

2251      N2 = 8
2252      L = 0
2253      IF (ID .GT. 0) N1 = 1
2254      DO 490 N = N1, N2
2255          L = L + 1
2256          IN(J,L) = NODE1(N)
2257      490 CONTINUE
2258      IF (ID .LT. 1) GO TO 500
2259      500 CONTINUE
2260      IFE(5) = IFE(5) + 1
2261      IFACE(IFE(5),5) = J
2262      IFE(4) = IFE(4) + 1
2263      IFACE(IFE(4),4) = J
2264      IF (IL .GT. 1 .OR. IC1 .GT. 1) GO TO 510
2265      IFE(2) = IFE(2) + 1
2266      IFACE(IFE(2),2) = J
2267      510 IF (IL .LT. NZSTPS) GO TO 520
2268      IFE(1) = IFE(1) + 1
2269      IFACE(IFE(1),1) = J
2270      J = J + 1
2271      IF (ID .GT. 0) NDDO = NDZ(NDDO)
2272      530 IF (ID .GT. 0 .AND. IL .LT. NZSTPS) GO TO 10
2273      NELM = J - 1
2274      NPTS = NPTS + I - 1
2275      540 FORMAT (3I5)
2276      DO 580 L1 = 1, NPTS
2277          DO 550 L2 = 1, NPTS
2278              IF (L1 .EQ. NDX(L2)) GO TO 580
2279          550 CONTINUE
2280          DO 560 L3 = 1, NPTS
2281              IF (L1 .EQ. NDY(L3)) GO TO 580
2282          560 CONTINUE
2283          IF (ID .LT. 1) GO TO 590
2284          DO 570 L4 = 1, NPTS
2285              IF (L1 .EQ. NDZ(L4)) GO TO 580
2286          570 CONTINUE
2287          GO TO 590
2288      580 CONTINUE
2289      GO TO 600
2290      590 IK = L1
2291      600 CONTINUE
2292      DELX = ABS(DELX)
2293      DELY = ABS(DELY)
2294      RETURN
2295      END
2296      C
2297      C
2298      C .....
2299      C Subroutine ELMNT
2300      C

```



```

2301 C
2302 C
2303 C      Author
2304 C      G.J. Smith(U. of Alberta)
2305 C
2306 C      Last Update
2307 C      May 1984
2308 C
2309 C      Purpose
2310 C      To simultaneously form and number the elements
2311 C      while the nodes are being generated.
2312 C
2313 C      Usage
2314 C      CALL ELMNT(I,J,NPTSI,IROW,ICOL,ND1,NDD2,NDD3,IL,IC1,IFLG)
2315 C
2316 C      Subroutines Required
2317 C      None
2318 C
2319 C      .....
2320 C      SUBROUTINE ELMNT(I,J,NPTSI,IROW,ICOL,ND1,NDD2,NDD3,IL,
2321 C      1 IC1,IFLG)
2322 C      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
2323 C      1 ZN(5000)
2324 C      COMMON CX(5000)
2325 C      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
2326 C      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
2327 C      1 ALEAK(100)
2328 C      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
2329 C      1 PPOSZ(10)
2330 C      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
2331 C      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
2332 C      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
2333 C      COMMON IP, NPTS, NPTSI, NELM, IO, ITRNS, INC, IELEM, NZSTPS, ID,
2334 C      1 YC, XC, IPN
2335 C      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
2336 C      1 DELTZ
2337 C      NXSTP1 = NXSTPS - 1
2338 C      NYSTP1 = NYSTPS - 1
2339 C      IF (ID.GT. 0) NZSTP1 = NZSTPS - 1
2340 C      IF (ICOL.EQ. NXSTPS) GO TO 120
2341 C      IF (ID.GT. 0 .OR. IELEM.EQ. 1) GO TO 20
2342 C      IF (IFLG.EQ. 1) GO TO 10
2343 C 2-D Triangular elements
2344 C      IN(J,1) = NODE1(5)
2345 C      IN(J,2) = NODE1(6)
2346 C      IN(J,3) = NODE1(8)
2347 C      J = J + 1
2348 C      IN(J,1) = NODE1(8)
2349 C      IN(J,2) = NODE1(6)
2350 C      IN(J,3) = NDX(NODE1(6))
2351 C      NDX(NODE1(5)) = IN(J,1)

```



```

2351 J = J + 1
2352 NODE1(5) = NPTSI
2353 NODE1(6) = IN(J - 1,3)
2354 IFLG = 1
2355 RETURN
2356 10 IN(J,1) = NODE1(5)
2357 IN(J,2) = NODE1(6)
2358 IN(J,3) = NDX(NODE1(6))
2359 J = J + 1
2360 IN(J,1) = NODE1(5)
2361 IN(J,2) = NDX(NODE1(6))
2362 IN(J,3) = NODE1(8)
2363 NDX(NODE1(5)) = IN(J,3)
2364 J = J + 1
2365 NODE1(5) = NPTSI
2366 NODE1(6) = IN(J - 1,2)
2367 NODE1(7) = NDX(NODE1(6))
2368 IFLG = 0
2369 RETURN
2370 20 IF (IELEM .LT. 1) GO TO 100
2371 N1 = 5
2372 N2 = 8
2373 L = 0
2374 IF (ID .GT. 0) N1 = 1
2375 DO 30 N = N1, N2
2376 L = L + 1
2377 IN(J,L) = NODE1(N)
2378 30 CONTINUE
2379 IF (ID .LT. 1) GO TO 40
2380 40 CONTINUE
2381 50 FORMAT (9I5)
2382 60 FORMAT (4I5)
2383 70 FORMAT (15, 3F13.3)
2384 80 FORMAT (2I5)
2385 J = J + 1
2386 C Tetrahedral elements
2387 IF (ID .LT. 1) GO TO 90
2388 NODE1(1) = NODE1(4)
2389 NODE1(2) = NODE1(3)
2390 90 NODE1(5) = NODE1(8)
2391 NODE1(6) = NODE1(7)
2392 RETURN
2393 100 IF (IFLG .EQ. 1) GO TO 110
2394 IN(J,1) = NODE1(1)
2395 IN(J,2) = NODE1(2)
2396 IN(J,3) = NODE1(4)
2397 IN(J,4) = NODE1(5)
2398 J = J + 1
2399 IN(J,1) = NODE1(2)
2400 IN(J,2) = NODE1(7)

```



```

2401 IN(J,3) = NODE1(5)
2402 IN(J,4) = NODE1(6)
2403 J = J + 1
2404 IN(J,1) = NODE1(2)
2405 IN(J,2) = NODE1(3)
2406 IN(J,3) = NODE1(4)
2407 IN(J,4) = NODE1(7)
2408 J = J + 1
2409 IN(J,1) = NODE1(4)
2410 IN(J,2) = NODE1(5)
2411 IN(J,3) = NODE1(7)
2412 IN(J,4) = NODE1(8)
2413 J = J + 1
2414 IN(J,1) = NODE1(4)
2415 IN(J,2) = NODE1(2)
2416 IN(J,3) = NODE1(7)
2417 IN(J,4) = NODE1(5)
2418 J = J + 1
2419 IFLG = 1
2420 NODE1(1) = NODE1(4)
2421 NODE1(2) = NODE1(3)
2422 NODE1(5) = NODE1(8)
2423 NODE1(6) = NODE1(7)
2424 RETURN
110 IN(J,1) = NODE1(1)
2425 IN(J,2) = NODE1(2)
2426 IN(J,3) = NODE1(3)
2427 IN(J,4) = NODE1(6)
2428 J = J + 1
2429 IN(J,1) = NODE1(3)
2430 IN(J,2) = NODE1(6)
2431 IN(J,3) = NODE1(7)
2432 IN(J,4) = NODE1(8)
2433 J = J + 1
2434 IN(J,1) = NODE1(3)
2435 IN(J,2) = NODE1(4)
2436 IN(J,3) = NODE1(1)
2437 IN(J,4) = NODE1(8)
2438 J = J + 1
2439 IN(J,1) = NODE1(1)
2440 IN(J,2) = NODE1(6)
2441 IN(J,3) = NODE1(8)
2442 IN(J,4) = NODE1(5)
2443 J = J + 1
2444 IN(J,1) = NODE1(3)
2445 IN(J,2) = NODE1(1)
2446 IN(J,3) = NODE1(6)
2447 IN(J,4) = NODE1(8)
2448 J = J + 1
2449 IFLG = 0
2450

```



```

2451 NODE1(1) = NODE1(4)
2452 NODE1(2) = NODE1(3)
2453 NODE1(5) = NODE1(8)
2454 NODE1(6) = NODE1(7)
2455 RETURN
2456
2457 120 CONTINUE
2458 IF (IELEM .GT. 0 .OR. ID .GT. 0) GO TO 140
2459 IF (IFLG .EQ. 1) GO TO 130
2460 IN(J,1) = NODE1(5)
2461 IN(J,2) = NODE1(6)
2462 IN(J,3) = ICRNR(IC1,2) + IROW
2463 IF (ICRNR(IC1,2) .EQ. 1) IN(J,3) = ICRNR(IC1,3) + NYSTP1 - (IROW -
1 1)
2464 J = J + 1
2465 IN(J,1) = ICRNR(IC1,2) + IROW
2466 IF (ICRNR(IC1,2) .EQ. 1) IN(J,1) = ICRNR(IC1,3) + NYSTP1 - (IROW -
1 1)
2467 IN(J,2) = NODE1(6)
2468 IN(J,3) = ICRNR(IC1,2) + IROW - 1
2469 IF (ICRNR(IC1,2) .EQ. 1) IN(J,3) = ICRNR(IC1,3) + NYSTP1 - (IROW -
1 2)
2470 IF (IROW .EQ. 1) IN(J,3) = ICRNR(IC1,2)
2471 NDX(IN(J - 1,1)) = IN(J,1)
2472 NODE1(8) = IN(J,1)
2473 J = J + 1
2474 IFLG = 1
2475 GO TO 230
2476
2477 130 IN(J,1) = NODE1(5)
2478 IN(J,2) = NODE1(6)
2479 IN(J,3) = ICRNR(IC1,2) + IROW - 1
2480 IF (ICRNR(IC1,2) .EQ. 1) IN(J,3) = ICRNR(IC1,3) + NYSTP1 - (IROW -
1 2)
2481 IF (IROW .EQ. 1) IN(J,3) = ICRNR(IC1,2)
2482 J = J + 1
2483 IN(J,1) = NODE1(5)
2484 IN(J,2) = ICRNR(IC1,2) + IROW - 1
2485 IF (ICRNR(IC1,2) .EQ. 1) IN(J,2) = ICRNR(IC1,3) + NYSTP1 - (IROW -
1 2)
2486 IF (IROW .EQ. 1) IN(J,2) = ICRNR(IC1,2)
2487 IN(J,3) = ICRNR(IC1,2) + IROW
2488 IF (ICRNR(IC1,2) .EQ. 1) IN(J,3) = ICRNR(IC1,3) + NYSTP1 - (IROW -
1 1)
2489 NDX(IN(J,1)) = IN(J,3)
2490 NODE1(8) = IN(J,3)
2491 J = J + 1
2492 IFLG = 0
2493 GO TO 230
2494
2495 C Quadrilateral and parallelepiped elements
2496 140 NODE1(7) = NDX(NODE1(6))
2497 NODE1(8) = NDY(NODE1(7))
2500

```



```

2501 NDX(NODE1(5)) = NODE1(8)
2502 NDY(NODE1(7)) = NODE1(8)
2503 IF (ID .LT. 1) GO TO 200
2504 NODE1(3) = NDZ(NODE1(7))
2505 NDX(NODE1(2)) = NODE1(3)
2506 IFE(4) = IFE(4) + 1
2507 IFACE(IFE(4),4) = J
2508 IF (IL .EQ. NZSTPS .AND. ID .GT. 0) GO TO 150
2509 GO TO 170
2510
2510 NODE1(4) = NDY(NODE1(3))
2511 NDX(NODE1(1)) = NODE1(4)
2512 NDZ(NODE1(8)) = NODE1(4)
2513 IFE(1) = IFE(1) + 1
2514 IFACE(IFE(1),1) = J
2515 IF (IROW .GT. 1) GO TO 160
2516 IFE(3) = IFE(3) + 1
2517 IFACE(IFE(3),3) = J
2518 IF (IL .GT. 1 .OR. IC1 .GT. 1) GO TO 200
2519 IFE(2) = IFE(2) + 1
2520 IFACE(IFE(2),2) = J
2521 GO TO 200
2522
2522 CONTINUE
2523 XN(NPTS + I) = XN(NODE1(8))
2524 YN(NPTS + I) = YN(NODE1(8))
2525 ZN(NPTS + I) = ZN(NODE1(8)) + DELTZ
2526 NODE1(4) = NPTS + I
2527 IF (ID .LT. 1) CX(NPTS + I) = COX(ITRNS)
2528 NDY(NODE1(3)) = NPTS + I
2529 NDZ(NODE1(8)) = NPTS + I
2530 NDX(NODE1(1)) = NODE1(4)
2531 NBN = NBN + 1
2532 NBDND(NBN) = NPTS + I
2533 IFE(4) = IFE(4) + 1
2534 IFACE(IFE(4),4) = J
2535 IF (IL .GT. 1 .OR. IC1 .GT. 1) GO TO 180
2536 IFE(2) = IFE(2) + 1
2537 IFACE(IFE(2),2) = J
2538 IF (IROW .GT. 1) GO TO 190
2539 IFE(3) = IFE(3) + 1
2540 IFACE(IFE(3),3) = J
2541 CONTINUE
2542 I = I + 1
2543 IF (IELEM .LT. 1) GO TO 250
2544 N1 = 5
2545 N2 = 8
2546 L = 0
2547 IF (ID .GT. 0) N1 = 1
2548 DO 210 N = N1, N2
2549   L = L + 1
2550   IN(J,L) = NODE1(N)

```



```

2551 210 CONTINUE
2552 IF (ID.LT. 1) GO TO 220
2553 220 CONTINUE
2554 J = J + 1
2555 230 CONTINUE
2556 240 FORMAT (I5)
2557 GO TO 270
2558 250 IF (IFLG.EQ. 1) GO TO 260
2559 C Tetrahedral elements
2560 IN(J,1) = NODE1(1)
2561 IN(J,2) = NODE1(2)
2562 IN(J,3) = NODE1(4)
2563 IN(J,4) = NODE1(5)
2564 J = J + 1
2565 IN(J,1) = NODE1(2)
2566 IN(J,2) = NODE1(7)
2567 IN(J,3) = NODE1(5)
2568 IN(J,4) = NODE1(6)
2569 J = J + 1
2570 IN(J,1) = NODE1(2)
2571 IN(J,2) = NODE1(3)
2572 IN(J,3) = NODE1(4)
2573 IN(J,4) = NODE1(7)
2574 J = J + 1
2575 IN(J,1) = NODE1(4)
2576 IN(J,2) = NODE1(5)
2577 IN(J,3) = NODE1(7)
2578 IN(J,4) = NODE1(8)
2579 J = J + 1
2580 IN(J,1) = NODE1(4)
2581 IN(J,2) = NODE1(2)
2582 IN(J,3) = NODE1(7)
2583 IN(J,4) = NODE1(5)
2584 J = J + 1
2585 IFLG = 1
2586 NODE1(1) = NODE1(4)
2587 NODE1(2) = NODE1(3)
2588 NODE1(5) = NODE1(8)
2589 NODE1(6) = NODE1(7)
2590 GO TO 270
2591 260 IN(J,1) = NODE1(1)
2592 IN(J,2) = NODE1(2)
2593 IN(J,3) = NODE1(3)
2594 IN(J,4) = NODE1(6)
2595 J = J + 1
2596 IN(J,1) = NODE1(3)
2597 IN(J,2) = NODE1(6)
2598 IN(J,3) = NODE1(7)
2599 IN(J,4) = NODE1(8)
2600 J = J + 1

```



```

2601 IN(J,1) = NODE1(3)
2602 IN(J,2) = NODE1(4)
2603 IN(J,3) = NODE1(1)
2604 IN(J,4) = NODE1(8)
2605 J = J + 1
2606 IN(J,1) = NODE1(1)
2607 IN(J,2) = NODE1(6)
2608 IN(J,3) = NODE1(8)
2609 IN(J,4) = NODE1(5)
2610 J = J + 1
2611 IN(J,1) = NODE1(3)
2612 IN(J,2) = NODE1(1)
2613 IN(J,3) = NODE1(6)
2614 IN(J,4) = NODE1(8)
2615 J = J + 1
2616 IFLG = 0
2617 270 NDD2 = NDY(NODE1(8))
2618 ND1 = NDD3
2619 RETURN
2620 END
2621
2622
2623
2624
2625 Subroutine SETUP
2626
2627 Author
2628 R.J. Collins 1973, modified G.J. Smith 1984
2629
2630 Last Update
2631 May 1984
2632
2633 Purpose
2634 To set up the interconnection of the nodes
2635 in such a way as to make it easy to
2636 renumber to reduce bandwidth.
2637
2638 Usage
2639 CALL SETUP(IDIFF)
2640
2641 Subroutines Required
2642 None
2643
2644
2645
2646 SUBROUTINE SETUP(IDIFF)
2647 COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
2648 1 ZN(5000)
2649 COMMON CX(5000)
2650 COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)

```



```

2651 COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
2652 1 ALEAK(100)
2653 COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
2654 1 PPOSZ(10)
2655 COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
2656 COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
2657 COMMON ICRNR(10,8), NFACE(6,4), ICONTST(4,2), NODE1(8)
2658 COMMON IP, NODES, NPTS1, LMENTS, IO, ITRNS, INC, IELEM, NZ, ID,
2659 1 YC, XC, IPN
2660 COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
2661 1 DELTZ
2662 DIMENSION JNT(5000), NDD(3)
2663 IDIFF = NODES
2664 10 FORMAT ('NODES=', I5)
2665 DO 30 J = 1, NODES
2666 JMEM(J) = 0
2667 DO 20 I = 1, 32
2668 MEMJT(J,I) = 0
2669 20 CONTINUE
2670 30 CONTINUE
2671 C Scan through the elements to form an array
2672 C indicating which nodes are interconnected.
2673 DO 80 J = 1, LMENTS
2674 DO 70 I = 1, INC
2675 JNTI = IN(J,I)
2676 IF (JNTI .EQ. 0) GO TO 80
2677 DO 60 II = 1, INC
2678 IF (II .EQ. I) GO TO 60
2679 JJT = IN(J,II)
2680 IF (JNTI .EQ. JJT) GO TO 60
2681 MEM1 = JMEM(JNTI)
2682 IF (MEM1 .EQ. 0) GO TO 50
2683 DO 40 III = 1, MEM1
2684 IF (MEMJT(JNTI,III) .EQ. JJT) GO TO 60
2685 40 CONTINUE
2686 JMEM(JNTI) = JMEM(JNTI) + 1
2687 MEMJT(JNTI,JMEM(JNTI)) = JJT
2688 IF (IABS(JNTI - JJT) .GT. IDIFF) IDIFF = IABS(JNTI - JJT)
2689 50 CONTINUE
2690 60 CONTINUE
2691 70 CONTINUE
2692 80 CONTINUE
2693 DO 100 J = 1, NODES
2694 MEM1 = JMEM(J)
2695 90 FORMAT (9I5)
2696 100 CONTINUE
2697 IDIFF = IDIFF + 1
2698 RETURN
2699 END

```

C
C


```

2701 C .....
2702 C
2703 C Subroutine OPTNUM
2704 C
2705 C Author
2706 C R.J. Collins 1973, modified G.J. Smith 1984
2707 C
2708 C Last Update
2709 C May 1984
2710 C
2711 C Purpose
2712 C To renumber the nodes of the mesh in such
2713 C a way as to reduce bandwidth and hence
2714 C increase computational efficiency.
2715 C
2716 C Usage
2717 C CALL OPTNUM(IDIFF,MAX)
2718 C
2719 C Subroutines Required
2720 C None
2721 C
2722 C
2723 C .....
2724 C SUBROUTINE OPTNUM(IDIFF, MAX)
2725 C COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
2726 C 1 ZN(5000)
2727 C COMMON CX(5000)
2728 C COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
2729 C COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
2730 C 1 ALEAK(100)
2731 C COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
2732 C 1 PPOSZ(10)
2733 C COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
2734 C COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
2735 C COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
2736 C COMMON IP, NODES, NPTS1, LMENTS, IO, ITRNS, INC, IELEM, NZ, ID,
2737 C 1 YC, XC, IPN
2738 C COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
2739 C 1 DELTZ
2740 C DIMENSION NEWIN(3000,8), XNN(5000), YNN(5000), ZNN(5000),
2741 C 1 JOINT(5000)
2742 C DIMENSION NEWCX(5000), NTYPE(5000)
2743 C DIMENSION JNT(5000), NEWJT(5000)
2744 C MINMAX = IDIFF
2745 C NPN = 0
2746 C 10 FORMAT ('NPTS=', I5)
2747 C DO 30 J = 1, NODES
2748 C MEM1 = JMEM(J)
2749 C 20 FORMAT (9I5)
2750 C JNT(J) = 0

```



```

2751 30 CONTINUE
2752 DO 40 J = 1, NODES
2753   JOINT(J) = 0
2754   JNT(J) = 0
2755   NTYPE(J) = 0
2756 40 NEWJT(J) = 0
2757   MAX = 0
2758   I = 1
2759   NEWJT(1) = IK
2760   JOINT(IK) = 1
2761   K = 1
2762 50 K4 = JMEM(NEWJT(I))
2763   IF (K4 .EQ. 0) GO TO 80
2764 60 FORMAT (2I5)
2765   DO 70 JJ = 1, K4
2766     K5 = MEMJT(NEWJT(I),JJ)
2767     IF (JOINT(K5) .GT. 0) GO TO 70
2768     K = K + 1
2769     NEWJT(K) = K5
2770     JOINT(K5) = K
2771     NDIFF = IABS(I - K)
2772     IF (NDIFF .GE. MINMAX) GO TO 110
2773     IF (NDIFF .GT. MAX) MAX = NDIFF
2774 70 CONTINUE
2775     IF (K .GE. NODES) GO TO 90
2776     I = I + 1
2777     GO TO 50
2778 90 MINMAX = MAX
2779     DO 100 J = 1, NODES
2780       JNT(J) = JOINT(J)
2781 110 CONTINUE
2782     IF (JNT(1) .EQ. 0) GO TO 200
2783     DO 140 I = 1, NODES
2784       ND1 = NEWJT(I)
2785       XNN(I) = XN(ND1)
2786       YNN(I) = YN(ND1)
2787       ZNN(I) = ZN(ND1)
2788       NTYPE(I) = ITYPE(ND1)
2789       IF (ITYPE(ND1) .NE. 2) GO TO 120
2790       NPN = NPN + 1
2791       NPMP(NPN) = I
2792 120 IF (ID .LT. 1) NEWCX(I) = CX(ND1)
2793 130 FORMAT (15, 2F10.3)
2794 140 CONTINUE
2795     DO 160 I = 1, LMENTS
2796       DO 150 J = 1, INC
2797         JND = IN(I,J)
2798         NEWIN(I,J) = JOINT(JND)
2799 150 CONTINUE
2800 160 CONTINUE

```



```

2801      DO 170 I = 1, NODES
2802         XN(I) = XNN(I)
2803         YN(I) = YNN(I)
2804         ZN(I) = ZNN(I)
2805         ITYPE(I) = NTYPE(I)
2806         IF (ID .LT. 1) CX(I) = NEWCX(I)
2807
2808         170 CONTINUE
2809         DO 190 I = 1, LMENTS
2810            DO 180 J = 1, INC
2811               IN(I,J) = NEWIN(I,J)
2812            180 CONTINUE
2813            190 CONTINUE
2814            200 CONTINUE
2815            MAX = MAX + 1
2816            RETURN
2817            END
2818
2819      C .....
2820      C Subroutine NDEGEN
2821
2822      C Author
2823      C      G.J. Smith
2824
2825      C Last Update
2826      C      May 1984
2827
2828      C Purpose
2829      C      This subroutine generates a node, given a
2830      C      radial distance and an azimuth from a
2831      C      reference node.
2832
2833      C Usage
2834      C      CALL NDEGEN(I,II,PHI,R,THETA,Ndd)
2835
2836      C Subroutines Required
2837
2838      C      None
2839
2840      C .....
2841      C .....
2842      C SUBROUTINE NDEGEN(I, II, PHI, R, THETA, NDD)
2843      C COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
2844      C      1 ZN(5000)
2845      C COMMON CX(5000)
2846      C COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
2847      C COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
2848      C      1 ALEAK(100)
2849      C COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
2850      C      1 PPOSZ(10)

```



```

2851 COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
2852 COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
2853 COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
2854 COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
2855 1 XC, IPN
2856 COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
2857 1 DELTZ
2858 10 FORMAT (3F13.3)
2859 PHI = PHI + THETA
2860 IF (PHI .GT. 6.283) PHI = PHI - 6.283
2861 XA1 = COS(PHI) * R
2862 YA1 = SIN(PHI) * R
2863 20 FORMAT (2F13.3)
2864 YN(NPTS + II) = YN(I) + YA1
2865 XN(NPTS + II) = XN(I) + XA1
2866 IF (ID .GT. 0) ZN(NPTS + II) = ZN(I)
2867 NDD = NPTS + II
2868 RETURN
2869 END
2870
2871 .....
2872
2873 Subroutine INTANG
2874
2875 Author
2876 G.J. Smith
2877
2878 Last Update
2879 May 1984
2880
2881 Purpose
2882 Calculates the interior angle(azimuth) from
2883 the horizontal to the required node from
2884 a reference node.
2885
2886 Usage
2887 CALL INTANG(ANGLE,I,TEMP,I2,THETA)
2888
2889 Subroutines Required
2890 None
2891
2892 .....
2893
2894 SUBROUTINE INTANG(ANGLE, I, TEMP, I2, THETA)
2895 COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
2896 1 ZN(5000)
2897 COMMON CX(5000)
2898 COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
2899 COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
2900

```



```

2901 1 ALEAK(100)
2902 COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
2903 1 PPOSZ(10)
2904 COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
2905 COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
2906 COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
2907 COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
2908 1 XC, IPN
2909 COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
2910 1 DELTZ
2911 DX1 = TEMP
2912 X1 = ABS(DX1)
2913 DX2 = XN(I) - XN(I2)
2914 X2 = ABS(DX2)
2915 DY1 = O.OO
2916 Y1 = ABS(DY1)
2917 DY2 = YN(I) - YN(I2)
2918 Y2 = ABS(DY2)
2919
2920 C Since some systems do not allow negative arguments in the tangent
2921 C function,
2922 C one must orientate the angle otherwise.
2923 IF (DX1.GT. 0.OO .AND. DY1.GE. 0.OO) THETA1 = ATAN2(Y1,X1)
2924 IF (DX1.LT. 0.OO .AND. DY1.GT. 0.OO) THETA1 = 3.1412 - ATAN2(Y1,
2925 1X1)
2926 IF (DX1.LT. 0.OO .AND. DY1.LT. 0.OO) THETA1 = 3.1412 + ATAN2(Y1,
2927 1X1)
2928 IF (DX1.GT. 0.OO .AND. DY1.LT. 0.OO) THETA1 = 6.282 - ATAN2(Y1,
2929 1X1)
2930 IF (DX2.LT. 0.OO .AND. DY2.LT. 0.OO) THETA2 = ATAN2(Y2,X2)
2931 IF (DX2.GT. 0.OO .AND. DY2.LT. 0.OO) THETA2 = 3.1412 - ATAN2(Y2,
2932 1X2)
2933 IF (DX2.GT. 0.OO .AND. DY2.GT. 0.OO) THETA2 = 3.1412 + ATAN2(Y2,
2934 1X2)
2935 IF (DX2.LT. 0.OO .AND. DY2.GT. 0.OO) THETA2 = 6.282 - ATAN2(Y2,
2936 1X2)
2937 ANGLE = (THETA2 - THETA1)
2938 IF (ANGLE.LT. 0.OO) ANGLE = 6.282 + ANGLE
2939 THETA = THETA1
2940 IF (DX1.GT. 0.OO .AND. X1.GT. Y1 .AND. DY1.LT. 0.OO .AND. DY2
2941 1.GT. 0.OO .AND. Y2.GT. Y1) ANGLE = 6.282 - ANGLE
2942 IF (DY1.GT. 0.OO .AND. Y1.GT. X1 .AND. DX1.GT. 0.OO .AND. DX2
2943 1.LT. 0.OO .AND. X1.LT. X2) ANGLE = 6.282 - ANGLE
2944 IF (DY1.GT. 0.OO .AND. Y1.GT. X1 .AND. DX1.GT. 0.OO .AND. DX2
2945 1.LT. 0.OO .AND. X2.LT. X1 .AND. DY.LT. 0.OO) ANGLE = 6.282 -
2946 2ANGLE
2947 RETURN
2948 END
2949
2950 C
2951 C
2952 C

```



```

2951 C
2952 C Subroutine ASKPLT
2953 C
2954 C Author
2955 C R. Buttuls 1979, modified G.J. Smith
2956 C
2957 C Last Update
2958 C May 1984
2959 C
2960 C Purpose
2961 C Queries and generates output for use on
2962 C the Calcomp plotter.
2963 C
2964 C Usage
2965 C CALL ASKPLT
2966 C
2967 C Subroutines Required
2968 C Integrated Graphics Library Subroutines
2969 C
2970 C
2971 C .....
2972 C SUBROUTINE ASKPLT
2973 C INTEGER YES /'Y'/
2974 C PRINT 10
2975 C 10 FORMAT ('&PLOT?')
2976 C READ (5,20) IANSWR
2977 C 20 FORMAT (A1)
2978 C IF (IANSWR.NE. YES) RETURN
2979 C WRITE (6,30)
2980 C 30 FORMAT ('Presently the size of the plot will fit on an', /,
2981 C 1 '8.5 x 11 inch sheet of paper. Dou you wish a ', /,
2982 C 2 'larger plot?')
2983 C READ (5,20) IANSWR
2984 C IF (IANSWR.NE. YES) GO TO 60
2985 C WRITE (6,40)
2986 C 40 FORMAT ('The screen is presently equal to 7.5 inches', /,
2987 C 1 'in both the horizontal and vertical directions', /,
2988 C 2 'Enter a value "S" in inches that you wish the ', /,
2989 C 3 'dimensions of the plotted information to cover.', /,
2990 C 4 'the format is F10.0.')
2991 C READ (5,50) S
2992 C S = S * 2.54
2993 C 50 FORMAT (F10.0)
2994 C *****
2995 C CALL IGCTRL('CALCOMP', 'SIZE', S)
2996 C *****
2997 C 60 CALL IGDRON('CALCOMP')
2998 C *****
2999 C PRINT 70
3000 C 70 FORMAT ('&TERMINATE PLOT?')

```



```

3001      READ (5,20) IANSWR
3002      IF (IANSWR.NE. YES) RETURN
3003      C *****
3004      CALL IGCIRL('CALCOMP', 'ENDPLOT')
3005      C *****
3006      RETURN
3007      END
3008
3009
3010
3011
3012      Subroutine CNCTQ
3013
3014      Author
3015          G.J. Smith
3016
3017      Last Update
3018          May 1984
3019
3020      Purpose
3021          Concentrates Quadrilateral and brick type
3022          elements around pumping wells.
3023
3024      Usage
3025          CALL CNCTQ(I,IM,IMM)
3026
3027      Subroutines Required
3028          None
3029
3030
3031
3032      C ..... SUBROUTINE CNCTQ(I, IM, IMM)
3033      COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
3034      1 ZN(5000)
3035      COMMON CX(5000)
3036      COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
3037      COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
3038      1 ALEAK(100)
3039      COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
3040      1 PPOSZ(10)
3041      COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
3042      COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
3043      COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
3044      COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
3045      1 XC, IPN
3046      COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
3047      1 DELTZ
3048      10 FORMAT (3I5)
3049      K = 1
3050      II = 1

```



```

3051 K1 = 1
3052 JELM = NELM + 1
3053 C Determine the interconnecting nodes and their relative position
3054 DO 40 J = K1, NELM
3055   DO 30 L = 1, INC
3056     IF (ID .GT. 0 .AND. K .GT. 8) GO TO 50
3057     IF (ID .LT. 1 .AND. K .GT. 4) GO TO 50
3058     IF (I .EQ. IN(J,L)) GO TO 20
3059     GO TO 30
3060   ISAVE(K,1) = J
3061   ISAVE(K,2) = L
3062   K = K + 1
3063   IF (ID .LT. 1) GO TO 40
3064   ISAVE(K,1) = J
3065   ISAVE(K,2) = L + 4
3066   K = K + 1
3067   GO TO 40
3068 30 CONTINUE
3069 40 CONTINUE
3070 50 KK = K - 1
3071   K1 = J
3072   TEMP = SQRT(DELX**2. + DELY**2.)
3073   R = 0.85 * TEMP
3074   R1 = 0.325 * TEMP
3075   K = 1
3076   DO 220 K1 = 1, KK
3077     IF (ID .LT. 1) CX(NPTS + II) = COX(ITRNS)
3078     IF (ID .LT. 1) CX(NPTS + II + 1) = COX(ITRNS)
3079     N1 = ISAVE(K1,1)
3080     N2 = ISAVE(K1,2)
3081     GO TO (60, 100, 140, 180, 60, 100, 140, 180), N2
3082     IF (N2 .GT. 4) GO TO 70
3083     NM1 = 1
3084     NM2 = 2
3085     NIN = 2
3086     IMM1 = IMM
3087     GO TO 80
3088   70 IF (IMM .GT. 2) GO TO 220
3089     NM1 = 9
3090     NM2 = 10
3091     NIN = 6
3092     IMM1 = IMM - 1
3093     IF (ID .LT. 1) IMM1 = IMM
3094   C form new nodes within elements surrounding the pumping well
3095   80 DO 90 NM = NM1, NM2
3096     C *****
3097     CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, IN(N1,NIN), THETA)
3098     C *****
3099     CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
3100     C *****

```



```

3101      II = II + 1
3102      *****
3103      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
3104      *****
3105      II = II + 1
3106      NIN = NIN + 1
3107      CONTINUE
3108      GO TO 220
3109      100 IF (N2.GT. 4) GO TO 110
3110      NM1 = 3
3111      NM2 = 4
3112      NIN = 3
3113      IMM1 = IMM
3114      GO TO 120
3115      110 IF (IMM.GT. 2) GO TO 220
3116      NM1 = 11
3117      NM2 = 12
3118      NIN = 7
3119      IMM1 = IMM - 1
3120      IF (ID.LT. 1) IMM1 = IMM
3121      DO 130 NM = NM1, NM2
3122      *****
3123      CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, IN(N1,NIN), THETA)
3124      *****
3125      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
3126      *****
3127      II = II + 1
3128      *****
3129      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
3130      *****
3131      II = II + 1
3132      NIN = NIN + 1
3133      CONTINUE
3134      GO TO 220
3135      140 IF (N2.GT. 4) GO TO 150
3136      NM1 = 5
3137      NM2 = 6
3138      NIN = 4
3139      IMM1 = IMM
3140      GO TO 160
3141      150 IF (IMM.GT. 2) GO TO 220
3142      NM1 = 13
3143      NM2 = 14
3144      NIN = 8
3145      IMM1 = IMM - 1
3146      IF (ID.LT. 1) IMM1 = IMM
3147      DO 170 NM = NM1, NM2
3148      *****
3149      CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, IN(N1,NIN), THETA)
3150      *****

```



```

3151      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
3152      II = II + 1
3153      *****
3154      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
3155      *****
3156      II = II + 1
3157      IF (N2.GT. 4) NIN = 5
3158      IF (N2.LT. 5) NIN = 1
3159      *****
3160      170 CONTINUE
3161      GO TO 220
3162      180 IF (N2.GT. 4) GO TO 190
3163      NM1 = 7
3164      NM2 = 8
3165      NIN = 1
3166      IMM1 = IMM
3167      GO TO 200
3168      190 IF (IMM.GT. 2) GO TO 220
3169      NM1 = 15
3170      NM2 = 16
3171      NIN = 5
3172      IMM1 = IMM - 1
3173      IF (ID.LT. 1) IMM1 = IMM
3174      200 DO 210 NM = NM1, NM2
3175      *****
3176      CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, IN(N1,NIN), THETA)
3177      *****
3178      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
3179      *****
3180      II = II + 1
3181      *****
3182      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
3183      *****
3184      NIN = NIN + 1
3185      II = II + 1
3186      210 CONTINUE
3187      220 CONTINUE
3188      IMM1 = IMM - 1
3189      IF (ID.LT. 1) IMM1 = IMM
3190      C form elements out of these newly generated nodes
3191      DO 250 NM = 1, 7
3192      IN(JELM,1) = ND(NM)
3193      IN(JELM,2) = NND(NM)
3194      IN(JELM,3) = NND(NM + 1)
3195      IN(JELM,4) = ND(NM + 1)
3196      IF (ID.LT. 1) GO TO 230
3197      IN(JELM,5) = ND(NM + 8)
3198      IN(JELM,6) = NND(NM + 8)
3199      IN(JELM,7) = NND(NM + 9)
3200      IN(JELM,8) = ND(NM + 9)

```



```

3201 NDZ(ND(NM + 8)) = ND(NM)
3202 NDZ(NND(NM + 8)) = NND(NM)
3203 NDZ(NND(NM + 9)) = NND(NM + 1)
3204 NDZ(ND(NM + 9)) = ND(NM + 1)
3205 CONTINUE
3206 FORMAT (9I5)
3207 JELM = JELM + 1
3208 CONTINUE
3209 IN(JELM,1) = ND(8)
3210 IN(JELM,2) = NND(8)
3211 IN(JELM,3) = NND(1)
3212 IN(JELM,4) = ND(1)
3213 IF (ID.LT. 1) GO TO 260
3214 IN(JELM,5) = ND(16)
3215 IN(JELM,6) = NND(16)
3216 IN(JELM,7) = NND(9)
3217 IN(JELM,8) = ND(9)
3218 NDZ(ND(16)) = ND(8)
3219 NDZ(NND(16)) = NND(8)
3220 CONTINUE
3221 JELM = JELM + 1
3222 IN(JELM,1) = NDWEL(IM,2,IMM)
3223 IN(JELM,2) = ND(1)
3224 IN(JELM,3) = ND(2)
3225 IN(JELM,4) = ND(3)
3226 IF (ID.LT. 1) GO TO 270
3227 IN(JELM,5) = NDWEL(IM,2,IMM1)
3228 IN(JELM,6) = ND(9)
3229 IN(JELM,7) = ND(10)
3230 IN(JELM,8) = ND(11)
3231 CONTINUE
3232 JELM = JELM + 1
3233 IN(JELM,1) = ND(5)
3234 IN(JELM,2) = NDWEL(IM,2,IMM)
3235 IN(JELM,3) = ND(3)
3236 IN(JELM,4) = ND(4)
3237 IF (ID.LT. 1) GO TO 280
3238 IN(JELM,5) = ND(13)
3239 IN(JELM,6) = NDWEL(IM,2,IMM1)
3240 IN(JELM,7) = ND(11)
3241 IN(JELM,8) = ND(12)
3242 CONTINUE
3243 JELM = JELM + 1
3244 IN(JELM,1) = ND(6)
3245 IN(JELM,2) = ND(7)
3246 IN(JELM,3) = NDWEL(IM,2,IMM)
3247 IN(JELM,4) = ND(5)
3248 IF (ID.LT. 1) GO TO 290
3249 IN(JELM,5) = ND(14)
3250 IN(JELM,6) = ND(15)

```



```

3251 IN(JELM,7) = NDWEL(IM,2,IMM1)
3252 IN(JELM,8) = ND(13)
3253
3254 JELM = JELM + 1
3255 IN(JELM,1) = ND(7)
3256 IN(JELM,2) = ND(8)
3257 IN(JELM,3) = ND(1)
3258 IN(JELM,4) = NDWEL(IM,2,IMM)
3259 IF (ID .LT. 1) GO TO 300
3260 IN(JELM,5) = ND(15)
3261 IN(JELM,6) = ND(16)
3262 IN(JELM,7) = ND(9)
3263 IN(JELM,8) = NDWEL(IM,2,IMM1)
3264
3265 JELM = JELM + 1
3266 DO 420 K1 = 1, KK
3267 N1 = ISAVE(K1,1)
3268 N2 = ISAVE(K1,2)
3269 IF (N2 .GT. 4) GO TO 420
3270 GO TO (310, 340, 370, 400), N2
3271 IN(JELM,1) = NND(3)
3272 IN(JELM,2) = NND(2)
3273 IN(JELM,3) = IN(N1,3)
3274 IN(JELM,4) = IN(N1,4)
3275 NDX(ND(1)) = ND(2)
3276 NDX(NND(1)) = NND(2)
3277 NDY(NND(2)) = NND(3)
3278 NDY(ND(2)) = ND(3)
3279 NDY(ND(1)) = NDWEL(IM,2,IMM)
3280 NDY(NND(1)) = ND(1)
3281 NDY(IN(N1,2)) = NND(1)
3282 NDX(ND(2)) = NND(2)
3283 NDX(NND(2)) = IN(N1,3)
3284 IF (ID .LT. 1) GO TO 320
3285 IN(JELM,5) = NND(11)
3286 IN(JELM,6) = NND(10)
3287 IN(JELM,7) = IN(N1,7)
3288 IN(JELM,8) = IN(N1,8)
3289 NDX(ND(9)) = ND(10)
3290 NDX(NND(1)) = NND(10)
3291 NDY(NND(10)) = NND(11)
3292 NDY(ND(10)) = ND(11)
3293 NDY(ND(9)) = NDWEL(IM,2,IMM1)
3294 NDY(NND(9)) = ND(9)
3295 NDY(IN(N1,6)) = NND(9)
3296 NDX(ND(10)) = NND(10)
3297 NDX(NND(10)) = IN(N1,7)
3298
3299 CONTINUE
3300 JELM = JELM + 1
3301 IN(N1,1) = NND(1)

```



```

3301 IN(N1,4) = NND(2)
3302 IF (ID .LT. 1) GO TO 330
3303 IN(N1,5) = NND(9)
3304 IN(N1,8) = NND(10)
3305 CONTINUE
3306 GO TO 420
3307 IN(JELM,1) = NND(4)
3308 IN(JELM,2) = NND(3)
3309 IN(JELM,3) = IN(N1,3)
3310 IN(JELM,4) = IN(N1,4)
3311 NDX(ND(5)) = ND(4)
3312 NDX(NND(5)) = ND(4)
3313 NDY(ND(5)) = NND(5)
3314 NDY(NND(5)) = IN(N1,1)
3315 NDY(ND(4)) = NND(4)
3316 NDY(NND(4)) = IN(N1,4)
3317 NDY(ND(3)) = ND(4)
3318 NDY(NND(3)) = NND(4)
3319 NDX(NDWEL(IM,2,IMM)) = ND(3)
3320 NDX(ND(3)) = NND(3)
3321 NDY(NDWEL(IM,2,IMM)) = ND(5)
3322 IF (ID .LT. 1) GO TO 350
3323 IN(JELM,5) = NND(12)
3324 IN(JELM,6) = NND(11)
3325 IN(JELM,7) = IN(N1,7)
3326 IN(JELM,8) = IN(N1,8)
3327 NDX(ND(13)) = ND(12)
3328 NDX(NND(13)) = ND(12)
3329 NDY(ND(13)) = NND(13)
3330 NDY(NND(13)) = IN(N1,5)
3331 NDY(ND(12)) = NND(12)
3332 NDY(NND(12)) = IN(N1,8)
3333 NDY(ND(11)) = ND(12)
3334 NDY(NND(11)) = NND(12)
3335 NDX(NDWEL(IM,2,IMM1)) = ND(11)
3336 NDX(ND(11)) = NND(11)
3337 NDY(NDWEL(IM,2,IMM1)) = ND(13)
3338 CONTINUE
3339 JELM = JELM + 1
3340 IN(N1,2) = NND(5)
3341 IN(N1,3) = NND(4)
3342 IF (ID .LT. 1) GO TO 360
3343 IN(N1,6) = NND(13)
3344 IN(N1,7) = NND(12)
3345 CONTINUE
3346 GO TO 420
3347 IN(JELM,1) = IN(N1,1)
3348 IN(JELM,2) = NND(6)
3349 IN(JELM,3) = NND(5)
3350 IN(JELM,4) = IN(N1,4)

```



```

3351 NDX(ND(6)) = ND(5)
3352 NDX(NND(6)) = NND(5)
3353 NDY(ND(6)) = NND(6)
3354 NDY(NND(6)) = IN(N1,1)
3355 NDX(NND(7)) = ND(7)
3356 NDX(IN(N1,2)) = NND(7)
3357 NDY(NND(7)) = NND(6)
3358 NDY(ND(7)) = ND(6)
3359 NDX(ND(7)) = NDWEL(IM,2,IMM)
3360 IF (ID.LT.1) GO TO 380
3361 IN(JELM,5) = IN(N1,5)
3362 IN(JELM,6) = NND(14)
3363 IN(JELM,7) = NND(13)
3364 IN(JELM,8) = IN(N1,8)
3365 NDX(ND(14)) = ND(13)
3366 NDX(NND(14)) = NND(13)
3367 NDY(ND(14)) = NND(14)
3368 NDY(NND(14)) = IN(N1,5)
3369 NDX(NND(15)) = ND(15)
3370 NDX(IN(N1,6)) = NND(15)
3371 NDY(NND(15)) = NND(14)
3372 NDY(ND(15)) = ND(14)
3373 NDX(ND(15)) = NDWEL(IM,2,IMM1)
3374
380 CONTINUE
3375 JELM = JELM + 1
3376 IN(N1,3) = NND(7)
3377 IN(N1,4) = NND(6)
3378 IF (ID.LT.1) GO TO 390
3379 IN(N1,7) = NND(15)
3380 IN(N1,8) = NND(14)
3381 CONTINUE
390
3382 GO TO 420
3383 IN(JELM,1) = IN(N1,1)
3384 IN(JELM,2) = IN(N1,2)
3385 IN(JELM,3) = NND(8)
3386 IN(JELM,4) = NND(7)
3387 NDX(ND(8)) = ND(1)
3388 NDX(NND(8)) = NND(1)
3389 NDY(NND(8)) = NND(7)
3390 NDY(ND(8)) = ND(7)
3391 NDY(IN(N1,2)) = NND(8)
3392 IF (ID.LT.1) GO TO 410
3393 IN(JELM,5) = IN(N1,5)
3394 IN(JELM,6) = IN(N1,6)
3395 IN(JELM,7) = NND(16)
3396 IN(JELM,8) = NND(15)
3397 NDX(ND(16)) = ND(9)
3398 NDX(NND(16)) = NND(9)
3399 NDY(NND(16)) = NND(15)
3400 NDY(ND(16)) = ND(15)

```



```

3401 NDY(IN(N1,6)) = NND(16)
3402 CONTINUE
3403 JELM = JELM + 1
3404 IN(N1,1) = NND(8)
3405 IN(N1,4) = NND(1)
3406 IF (ID.LT. 1) GO TO 420
3407 IN(N1,5) = NND(16)
3408 IN(N1,8) = NND(9)
3409 CONTINUE
3410 IF (ID.LT. 1) GO TO 440
3411 DO 430 I = 1, 8
3412 ND(I + 8) = ND(I)
3413 NND(I + 8) = NND(I)
3414 CONTINUE
3415 440 CONTINUE
3416 NPTS = NPTS + II - 1
3417 II = 1
3418 NELM = JELM - 1
3419 RETURN
3420 END
3421
3422
3423 .....
3424 Subroutine CNCTT1
3425
3426
3427 Author
3428 G.J. Smith
3429
3430 Last Update
3431 May 1984
3432
3433 Purpose
3434 To concentrate triangular and tetrahedral
3435 elements around a pumping well.
3436
3437 Usage
3438 CALL CNCTT1(I,IM,IMM)
3439
3440 Subroutines Required
3441 None
3442
3443
3444 .....
3445 SUBROUTINE CNCTT1(I, IM, IMM)
3446 COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
3447 1 ZN(5000)
3448 COMMON CX(5000)
3449 COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
3450 COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),

```



```

3451 1 ALEAK(100)
3452 COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
3453 1 PPOSZ(10)
3454 COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
3455 COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
3456 COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
3457 COMMON IP, NPTS, NPTS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
3458 1 XC, IPN
3459 COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
3460 1 DELTZ
3461 JELM = NELM + 1
3462 10 FORMAT (3I5)
3463 K = 1
3464 II = 1
3465 20 FORMAT (5I5)
3466 DO 50 J = 1, NELM
3467 DO 40 L = 1, INC
3468 IF (I.EQ. IN(J,L)) GO TO 30
3469 GO TO 40
3470 ISAVE(K,1) = J
3471 ISAVE(K,2) = L
3472 K = K + 1
3473 GO TO 50
3474 40 CONTINUE
3475 50 CONTINUE
3476 KK = K - 1
3477 TEMP = SQRT(DELX**2. + DELY**2.)
3478 R = 0.8 * TEMP
3479 R1 = 0.325 * TEMP
3480 K = 1
3481 IF (ID.LT. 1.AND. KK.GT. 4) GO TO 510
3482 DO 230 K1 = 1, KK
3483 IF (ID.LT. 1) CX(NPTS + II) = COX(ITRNS)
3484 IF (ID.LT. 1) CX(NPTS + II + 1) = COX(ITRNS)
3485 N1 = ISAVE(K1,1)
3486 N2 = ISAVE(K1,2)
3487 IF (ID.GT. 0) GO TO 60
3488 GO TO (70, 110, 150), N2
3489 IF (N2.NE. 3) GO TO 230
3490 IF (IN(N1,4).EQ. I.AND. IN(N1,1).EQ. I) GO TO 510
3491 IF (ABS(XN(IN(N1,3)) - XN(IN(N1,4))) .GT. ABS(YN(IN(N1,3)) - YN(
3492 IN(N1,4))) .A ND .(XN(IN(N1,4)) .GT. XN(IN(N1,3))) GO TO
3493 70
3494 IF (ABS(YN(IN(N1,3)) - YN(IN(N1,4))) .GT. ABS(XN(IN(N1,3)) - XN(
3495 IN(N1,4))) .A ND .(YN(IN(N1,4)) .GT. YN(IN(N1,3))) GO TO
3496 110
3497 IF (ABS(XN(IN(N1,3)) - XN(IN(N1,4))) .GT. ABS(YN(IN(N1,3)) - YN(
3498 IN(N1,4))) .A ND .(XN(IN(N1,4)) .LT. XN(IN(N1,3))) GO TO
3499 150
3500 IF (ABS(YN(IN(N1,3)) - YN(IN(N1,4))) .GT. ABS(XN(IN(N1,3)) - YN(

```



```

3501      1      IN(N1,4))) .) .A ND .(YN(IN(N1,4)) .LT. YN(IN(N1,3)))) GOTO
3502      2      190
3503
3504      70      NM1 = 1
3505             NM2 = 2
3506             NIN = IN(N1,4)
3507             IF (ID .LT. 1) NIN = IN(N1 + 1,2)
3508             IMM1 = IMM
3509             GO TO 90
3510
3511      80      IF (IMM .GT. 2) GO TO 230
3512             NM1 = 9
3513             NM2 = 10
3514             NIN = IN(N1,1)
3515             IMM1 = IMM - 1
3516             IF (ID .LT. 1) IMM1 = IMM
3517             DO 100 NM = NM1, NM2
3518             C *****
3519             CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
3520             C *****
3521             CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
3522             C *****
3523             II = II + 1
3524             C *****
3525             CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
3526             C *****
3527             II = II + 1
3528             NIN = NDX(NIN)
3529
3530      100     CONTINUE
3531             IF (ID .GT. 0 .AND. NM .LT. 4) GO TO 80
3532             GO TO 230
3533
3534      110     NM1 = 3
3535             NM2 = 4
3536             NIN = IN(N1,4)
3537             IF (ID .LT. 1) NIN = IN(N1,3)
3538             IMM1 = IMM
3539             GO TO 130
3540
3541      120     NM1 = 11
3542             NM2 = 12
3543             NIN = IN(N1,1)
3544             IMM1 = IMM - 1
3545             IF (ID .LT. 1) IMM1 = IMM
3546             DO 140 NM = NM1, NM2
3547             C *****
3548             CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
3549             C *****
3550             CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
3551             C *****
3552             II = II + 1
3553             C *****
3554             CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
3555             C *****

```



```

3551      II = II + 1
3552      NIN = NDY(NIN)
3553      CONTINUE
3554      IF (ID .GT. 0 .AND. NM .LT. 5) GO TO 120
3555      GO TO 230
3556      IF (ABS(YN(IN(N1,1))) - YN(IN(N1,3))) .LT. ABS(YN(IN(N1,2))) - YN(
3557      1 IN(N1,3))) .AND. ID .LT. 1) GO TO 190
3558      NM1 = 5
3559      NM2 = 6
3560      NIN = IN(N1,4)
3561      IF (ID .LT. 1) NIN = IN(N1 - 1,3)
3562      IMM1 = IMM
3563      GO TO 170
3564      IF (IMM .GT. 2) GO TO 230
3565      NM1 = 13
3566      NM2 = 14
3567      NIN = IN(N1,1)
3568      IMM1 = IMM - 1
3569      IF (ID .LT. 1) IMM1 = IMM
3570      DO 180 NM = NM1, NM2
3571      C *****
3572      CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
3573      C *****
3574      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
3575      C *****
3576      II = II + 1
3577      C *****
3578      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
3579      C *****
3580      II = II + 1
3581      NIN = IN(N1,2)
3582      IF (ID .LT. 1) NIN = IN(N1 - 1,2)
3583      IF (NM .GT. 7) NIN = NDZ(IN(N1,2))
3584      CONTINUE
3585      IF (ID .GT. 0 .AND. NM .LT. 7) GO TO 160
3586      GO TO 230
3587      NM1 = 7
3588      NM2 = 8
3589      NIN = IN(N1,4)
3590      IF (ID .LT. 1) NIN = IN(N1 - 1,1)
3591      IMM1 = IMM
3592      GO TO 210
3593      IF (IMM .GT. 2) GO TO 230
3594      NM1 = 15
3595      NM2 = 16
3596      NIN = IN(N1,1)
3597      IMM1 = IMM - 1
3598      IF (ID .LT. 1) IMM1 = IMM
3599      DO 220 NM = NM1, NM2
3600      C *****

```



```

3601      CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
3602      *****
3603      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
3604      *****
3605      II = II + 1
3606      *****
3607      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
3608      *****
3609      II = II + 1
3610      NIN = IN(N1,2)
3611      IF (ID .LT. 1) NIN = IN(N1 - 1,2)
3612      IF (NM .GT. 9) NIN = NDZ(IN(N1,2))
3613
3614      220 CONTINUE
3615      IF (ID .GT. 0 .AND. NM .LT. 9) GO TO 200
3616      230 CONTINUE
3617      IMM1 = IMM - 1
3618      IF (ID .LT. 1) IMM1 = IMM
3619      DO 280 NM = 1, 5, 2
3620      IN(JELM,1) = ND(NM)
3621      IN(JELM,2) = NND(NM)
3622      IN(JELM,3) = NND(NM + 1)
3623      IF (ID .LT. 1) GO TO 240
3624      IN(JELM,4) = ND(NM + 8)
3625      240 CONTINUE
3626      JELM = JELM + 1
3627      IN(JELM,1) = ND(NM)
3628      IN(JELM,2) = NND(NM + 1)
3629      IN(JELM,3) = ND(NM + 1)
3630      IF (ID .LT. 1) GO TO 250
3631      IN(JELM,4) = ND(NM + 9)
3632      250 CONTINUE
3633      JELM = JELM + 1
3634      IN(JELM,1) = ND(NM + 1)
3635      IN(JELM,2) = NND(NM + 1)
3636      IN(JELM,3) = ND(NM + 2)
3637      IF (ID .LT. 1) GO TO 260
3638      IN(JELM,4) = ND(NM + 9)
3639      260 CONTINUE
3640      JELM = JELM + 1
3641      IN(JELM,1) = ND(NM + 2)
3642      IN(JELM,2) = NND(NM + 1)
3643      IN(JELM,3) = NND(NM + 2)
3644      IF (ID .LT. 1) GO TO 270
3645      IN(JELM,4) = NND(NM + 10)
3646      270 CONTINUE
3647      JELM = JELM + 1
3648      IF (ID .LT. 1) GO TO 280
3649      IN(JELM,1) = NND(NM + 1)
3650      IN(JELM,2) = ND(NM)
3651      IN(JELM,3) = NND(NM + 8)

```



```

3651 IN(JELM,4) = ND(NM + 9)
3652 JELM = JELM + 1
3653 IN(JELM,1) = ND(NM)
3654 IN(JELM,2) = NND(NM + 3)
3655 IN(JELM,3) = ND(NM + 9)
3656 IN(JELM,4) = ND(NM + 8)
3657 JELM = JELM + 1
3658 IN(JELM,1) = NND(NM + 1)
3659 IN(JELM,2) = ND(NM + 9)
3660 IN(JELM,3) = NND(NM + 8)
3661 IN(JELM,4) = NND(NM + 9)
3662 JELM = JELM + 1
3663 IN(JELM,1) = NND(NM + 2)
3664 IN(JELM,2) = ND(NM + 1)
3665 IN(JELM,3) = NND(NM + 9)
3666 IN(JELM,4) = ND(NM + 10)
3667 JELM = JELM + 1
3668 IN(JELM,1) = NND(NM + 2)
3669 IN(JELM,2) = ND(NM + 10)
3670 IN(JELM,3) = NND(NM + 9)
3671 IN(JELM,4) = NND(NM + 10)
3672 JELM = JELM + 1
3673 IN(JELM,1) = ND(NM + 1)
3674 IN(JELM,2) = NND(NM + 10)
3675 IN(JELM,3) = ND(NM + 11)
3676 IN(JELM,4) = ND(NM + 10)
3677 JELM = JELM + 1
3678
280 CONTINUE
3679 IN(JELM,1) = ND(7)
3680 IN(JELM,2) = NND(7)
3681 IN(JELM,3) = NND(8)
3682 IF (ID .LT. 1) GO TO 290
3683 IN(JELM,4) = NND(15)
3684
290 CONTINUE
3685 JELM = JELM + 1
3686 IN(JELM,1) = ND(8)
3687 IN(JELM,2) = ND(7)
3688 IN(JELM,3) = NND(8)
3689 IF (ID .LT. 1) GO TO 300
3690 IN(JELM,4) = ND(16)
3691
300 CONTINUE
3692 JELM = JELM + 1
3693 IN(JELM,1) = ND(8)
3694 IN(JELM,2) = NND(8)
3695 IN(JELM,3) = ND(1)
3696 IF (ID .LT. 1) GO TO 310
3697 IN(JELM,4) = ND(16)
3698
310 CONTINUE
3699 JELM = JELM + 1
3700 IN(JELM,1) = ND(1)

```



```

3701 IN(JELM,2) = NND(8)
3702 IN(JELM,3) = NND(1)
3703 IF (ID .LT. 1) GO TO 320
3704 IN(JELM,4) = ND(9)
3705
320 CONTINUE
3706 JELM = JELM + 1
3707 IF (ID .LT. 1) GO TO 330
3708 IN(JELM,1) = NND(8)
3709 IN(JELM,2) = ND(7)
3710 IN(JELM,3) = NND(15)
3711 IN(JELM,4) = ND(16)
3712 JELM = JELM + 1
3713 IN(JELM,1) = NND(8)
3714 IN(JELM,2) = ND(16)
3715 IN(JELM,3) = NND(15)
3716 IN(JELM,4) = NND(16)
3717 JELM = JELM + 1
3718 IN(JELM,1) = ND(7)
3719 IN(JELM,2) = NND(15)
3720 IN(JELM,3) = ND(16)
3721 IN(JELM,4) = ND(15)
3722 JELM = JELM + 1
3723 IN(JELM,1) = NND(8)
3724 IN(JELM,2) = ND(1)
3725 IN(JELM,3) = ND(16)
3726 IN(JELM,4) = NND(9)
3727 JELM = JELM + 1
3728 IN(JELM,1) = ND(1)
3729 IN(JELM,2) = ND(16)
3730 IN(JELM,3) = NND(9)
3731 IN(JELM,4) = ND(9)
3732 JELM = JELM + 1
3733 IN(JELM,1) = NND(8)
3734 IN(JELM,2) = NND(9)
3735 IN(JELM,3) = ND(16)
3736 IN(JELM,4) = NND(16)
3737 JELM = JELM + 1
3738
330 CONTINUE
3739 DO 360 NM = 1, 5, 2
3740   IN(JELM,1) = NDWEL(IM,2,IMM)
3741   IN(JELM,2) = ND(NM)
3742   IN(JELM,3) = ND(NM + 2)
3743   IF (ID .LT. 1) GO TO 340
3744   IN(JELM,4) = NDWEL(IM,2,IMM1)
3745
340 CONTINUE
3746 JELM = JELM + 1
3747 IN(JELM,1) = ND(NM + 1)
3748 IN(JELM,2) = ND(NM + 2)
3749 IN(JELM,3) = ND(NM)
3750 IF (ID .LT. 1) GO TO 350

```



```

3751 IN(JELM,4) = ND(NM + 9)
3752 CONTINUE
3753 JELM = JELM + 1
3754 IF (ID .LT. 1) GO TO 360
3755 IN(JELM,1) = ND(NM)
3756 IN(JELM,2) = ND(NM + 2)
3757 IN(JELM,3) = NDWEL(IM,2,IMM1)
3758 IN(JELM,4) = NND(NM + 9)
3759 JELM = JELM + 1
3760 IN(JELM,1) = ND(NM)
3761 IN(JELM,2) = ND(NM + 9)
3762 IN(JELM,3) = NDWEL(IM,2,IMM1)
3763 IN(JELM,4) = ND(NM + 8)
3764 JELM = JELM + 1
3765 IN(JELM,1) = ND(NM + 2)
3766 IN(JELM,2) = NDWEL(IM,2,IMM1)
3767 IN(JELM,3) = ND(NM + 9)
3768 IN(JELM,4) = ND(NM + 10)
3769 JELM = JELM + 1
3770 CONTINUE
3771 IN(JELM,1) = NDWEL(IM,2,IMM)
3772 IN(JELM,2) = ND(7)
3773 IN(JELM,3) = ND(1)
3774 IF (ID .LT. 1) GO TO 370
3775 IN(JELM,4) = NDWEL(IM,2,IMM1)
3776 CONTINUE
3777 JELM = JELM + 1
3778 IN(JELM,1) = ND(8)
3779 IN(JELM,2) = ND(1)
3780 IN(JELM,3) = ND(7)
3781 IF (ID .LT. 1) GO TO 380
3782 IN(JELM,4) = ND(16)
3783 CONTINUE
3784 JELM = JELM + 1
3785 IF (ID .LT. 1) GO TO 390
3786 IN(JELM,1) = ND(7)
3787 IN(JELM,2) = ND(1)
3788 IN(JELM,3) = NDWEL(IM,2,IMM1)
3789 IN(JELM,4) = ND(16)
3790 JELM = JELM + 1
3791 IN(JELM,1) = ND(1)
3792 IN(JELM,2) = NDWEL(IM,2,IMM1)
3793 IN(JELM,3) = ND(16)
3794 IN(JELM,4) = ND(9)
3795 JELM = JELM + 1
3796 IN(JELM,1) = ND(7)
3797 IN(JELM,2) = ND(16)
3798 IN(JELM,3) = NDWEL(IM,2,IMM1)
3799 IN(JELM,4) = ND(15)
3800 JELM = JELM + 1

```



```

3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850

390 DO 490 K1 = 1, KK
    N1 = ISAVE(K1,1)
    N2 = ISAVE(K1,2)
    IF (ID.GT. 0) GO TO 400
    GO TO (410, 430, 450), N2
    IF (N2.NE. 3) GO TO 230
    IF (ABS(XN(IN(N1,3)) - XN(IN(N1,4)))) .GT. ABS(YN(IN(N1,3)) - YN(
1      IN(N1,4))) .) .A ND .(XN(IN(N1,4))) .GT. XN(IN(N1,3))) GOTO
2      410
    IF (ABS(YN(IN(N1,3)) - YN(IN(N1,4)))) .GT. ABS(XN(IN(N1,3)) - XN(
1      IN(N1,4))) .) .A ND .(YN(IN(N1,4))) .GT. YN(IN(N1,3))) GOTO
2      430
    IF (ABS(XN(IN(N1,3)) - XN(IN(N1,4)))) .GT. ABS(YN(IN(N1,3)) - YN(
1      IN(N1,4))) .) .A ND .(XN(IN(N1,4))) .LT. XN(IN(N1,3))) GOTO
2      450
    IF (ABS(YN(IN(N1,3)) - YN(IN(N1,4)))) .GT. ABS(XN(IN(N1,3)) - YN(
1      IN(N1,4))) .) .A ND .(YN(IN(N1,4))) .LT. YN(IN(N1,3))) GOTO
2      470
    IF (ID.GT. 0) GO TO 420
    IN(JELM,1) = IN(N1 + 1,3)
    IN(JELM,2) = NND(2)
    IN(JELM,3) = IN(N1,2)
    JELM = JELM + 1
    IN(JELM,1) = NND(3)
    IN(JELM,2) = NND(2)
    IN(JELM,3) = IN(N1 + 1,1)
    JELM = JELM + 1
    IN(N1,1) = NND(1)
    IN(N1,3) = NND(2)
    IN(N1 + 1,2) = NND(2)
    GO TO 490
    IN(JELM,1) = NND(3)
    IN(JELM,2) = NND(2)
    IN(JELM,3) = IN(N1 - 1,3)
    IN(JELM,4) = NND(11)
    JELM = JELM + 1
    IN(JELM,1) = NND(2)
    IN(JELM,2) = IN(N1,2)
    IN(JELM,3) = NND(11)
    IN(JELM,4) = NND(10)
    JELM = JELM + 1
    IN(JELM,1) = IN(N1 + 1,1)
    IN(JELM,2) = IN(N1 + 1,2)
    IN(JELM,3) = NND(2)
    IN(JELM,4) = IN(N1 + 1,4)
    JELM = JELM + 1
    IN(JELM,1) = IN(N1 + 2,1)
    IN(JELM,2) = NND(11)
    IN(JELM,3) = IN(N1 + 2,3)
    IN(JELM,4) = IN(N1 + 2,4)

```



```
3851 JELM = JELM + 1
3852 IN(JELM,1) = IN(N1 + 3,1)
3853 IN(JELM,2) = NND(2)
3854 IN(JELM,3) = IN(N1 + 3,3)
3855 IN(JELM,4) = NND(11)
3856 JELM = JELM + 1
3857 IN(N1 - 1,1) = NND(1)
3858 IN(N1 - 1,4) = NND(9)
3859 IN(N1,3) = NND(9)
3860 IN(N1 + 1,2) = NND(2)
3861 IN(N1 + 2,1) = NND(2)
3862 IN(N1 + 2,2) = NND(9)
3863 IN(N1 + 2,4) = NND(10)
3864 IN(N1 + 3,1) = NND(2)
3865 IN(N1 + 3,4) = NND(9)
3866 GO TO 490
3867 IF (ID .GT. O) GO TO 440
3868 IN(JELM,1) = NND(3)
3869 IN(JELM,2) = IN(N1,3)
3870 IN(JELM,3) = NND(4)
3871 JELM = JELM + 1
3872 IN(JELM,1) = NND(4)
3873 IN(JELM,2) = IN(N1,3)
3874 IN(JELM,3) = IN(N1 + 1,3)
3875 JELM = JELM + 1
3876 IN(N1 + 1,2) = NND(4)
3877 IN(N1,2) = NND(5)
3878 IN(N1,3) = NND(4)
3879 GO TO 490
3880 IN(JELM,1) = NND(5)
3881 IN(JELM,2) = NND(4)
3882 IN(JELM,3) = IN(N1 - 1,3)
3883 IN(JELM,4) = NND(13)
3884 JELM = JELM + 1
3885 IN(JELM,1) = NND(4)
3886 IN(JELM,2) = IN(N1,2)
3887 IN(JELM,3) = NND(13)
3888 IN(JELM,4) = NND(12)
3889 JELM = JELM + 1
3890 IN(JELM,1) = IN(N1 + 1,1)
3891 IN(JELM,2) = IN(N1 + 1,2)
3892 IN(JELM,3) = NND(4)
3893 IN(JELM,4) = IN(N1 + 1,4)
3894 JELM = JELM + 1
3895 IN(JELM,1) = IN(N1 + 2,1)
3896 IN(JELM,2) = NND(13)
3897 IN(JELM,3) = IN(N1 + 2,3)
3898 IN(JELM,4) = IN(N1 + 2,4)
3899 JELM = JELM + 1
3900 IN(JELM,1) = IN(N1 + 3,1)
```



```

3901 IN(JELM,2) = NND(4)
3902 IN(JELM,3) = IN(N1 + 3,3)
3903 IN(JELM,4) = NND(13)
3904 JELM = JELM + 1
3905 IN(N1 - 1,1) = NND(3)
3906 IN(N1 - 1,4) = NND(11)
3907 IN(N1,3) = NND(11)
3908 IN(N1 + 1,2) = NND(4)
3909 IN(N1 + 2,1) = NND(4)
3910 IN(N1 + 2,4) = NND(12)
3911 IN(N1 + 3,1) = NND(4)
3912 IN(N1 + 3,4) = NND(11)
3913 GO TO 490
3914 IF (ID.GT. O) GO TO 460
3915 IF (ABS(YN(IN(N1,1)) - YN(IN(N1,3)))) .LT. ABS(YN(IN(N1,2)) - YN(
1 IN(N1,3))) GO TO 470
3916 IN(JELM,1) = IN(N1,1)
3917 IN(JELM,2) = NND(6)
3918 IN(JELM,3) = NND(5)
3919 JELM = JELM + 1
3920 IN(JELM,1) = IN(N1 - 1,1)
3921 IN(JELM,2) = NND(6)
3922 IN(JELM,3) = IN(N1,1)
3923 JELM = JELM + 1
3924 IN(N1 - 1,3) = NND(6)
3925 IN(N1,1) = NND(6)
3926 IN(N1,3) = NND(7)
3927 GO TO 490
3928
3929 IN(JELM,1) = IN(N1 - 3,1)
3930 IN(JELM,2) = NND(6)
3931 IN(JELM,3) = IN(N1 - 3,3)
3932 IN(JELM,4) = IN(N1 - 3,4)
3933 JELM = JELM + 1
3934 IN(JELM,1) = NND(6)
3935 IN(JELM,2) = NND(13)
3936 IN(JELM,3) = IN(N1 - 2,3)
3937 IN(JELM,4) = NND(14)
3938 JELM = JELM + 1
3939 IN(JELM,1) = NND(5)
3940 IN(JELM,2) = IN(N1 - 1,2)
3941 IN(JELM,3) = NND(6)
3942 IN(JELM,4) = NND(13)
3943 JELM = JELM + 1
3944 IN(JELM,1) = IN(N1,1)
3945 IN(JELM,2) = IN(N1,2)
3946 IN(JELM,3) = NND(13)
3947 IN(JELM,4) = IN(N1,4)
3948 JELM = JELM + 1
3949 IN(JELM,1) = IN(N1 + 1,1)
3950 IN(JELM,2) = NND(6)

```



```

3951 IN(JELM,3) = NND(13)
3952 IN(JELM,4) = IN(N1 + 1,4)
3953 JELM = JELM + 1
3954 IN(N1 - 3,3) = NND(6)
3955 IN(N1 - 2,2) = NND(15)
3956 IN(N1 - 1,1) = NND(7)
3957 IN(N1 - 1,2) = NND(6)
3958 IN(N1 - 1,4) = NND(15)
3959 IN(N1,1) = NND(6)
3960 IN(N1,3) = NND(15)
3961 IN(N1,4) = NND(14)
3962 IN(N1 + 1,1) = NND(6)
3963 IN(N1 + 1,3) = NND(15)
3964 GO TO 490
470 IF (ID .GT. 0) GO TO 480
3965 IN(JELM,1) = IN(N1,1)
3966 IN(JELM,2) = IN(N1 - 1,2)
3967 IN(JELM,3) = NND(8)
3968 JELM = JELM + 1
3969 IN(JELM,1) = NND(8)
3970 IN(JELM,2) = IN(N1 - 1,2)
3971 IN(JELM,3) = IN(N1,2)
3972 JELM = JELM + 1
3973 IN(N1 - 1,1) = NND(1)
3974 IN(N1 - 1,2) = NND(8)
3975 IN(N1,2) = NND(8)
3976 IN(N1,3) = NND(7)
3977 GO TO 490
480 IN(JELM,1) = IN(N1 - 3,1)
3978 IN(JELM,2) = IN(N1 - 3,2)
3979 IN(JELM,3) = NND(8)
3980 IN(JELM,4) = IN(N1 - 3,4)
3981 JELM = JELM + 1
3982 IN(JELM,1) = IN(N1 - 2,1)
3983 IN(JELM,2) = NND(9)
3984 IN(JELM,3) = IN(N1 - 2,3)
3985 IN(JELM,4) = IN(N1 - 2,4)
3986 JELM = JELM + 1
3987 IN(JELM,1) = NND(1)
3988 IN(JELM,2) = NND(8)
3989 IN(JELM,3) = IN(N1 - 1,3)
3990 IN(JELM,4) = NND(9)
3991 JELM = JELM + 1
3992 IN(JELM,1) = NND(8)
3993 IN(JELM,2) = IN(N1,2)
3994 IN(JELM,3) = NND(9)
3995 IN(JELM,4) = NND(16)
3996 JELM = JELM + 1
3997 IN(JELM,1) = NND(8)
3998 IN(JELM,2) = IN(N1 + 1,2)
3999
4000

```



```

4001      IN(JELM,3) = NND(9)
4002      IN(JELM,4) = IN(N1 + 1,4)
4003      JELM = JELM + 1
4004      IN(N1 - 3,2) = NND(8)
4005      IN(N1 - 2,1) = NND(8)
4006      IN(N1 - 2,2) = NND(15)
4007      IN(N1 - 2,4) = NND(16)
4008      IN(N1 - 1,1) = NND(7)
4009      IN(N1 - 1,3) = NND(8)
4010      IN(N1 - 1,4) = NND(15)
4011      IN(N1,3) = NND(15)
4012      IN(N1 + 1,2) = NND(8)
4013      IN(N1 + 1,3) = NND(15)
4014      490 CONTINUE
4015      IF (ID .LT. 1) GO TO 520
4016      DO 500 I = 1, 8
4017          ND(I + 8) = ND(I)
4018          NND(I + 8) = NND(I)
4019      500 CONTINUE
4020      GO TO 520
4021      C *****
4022      510 CALL CNCTT2(I, IM, IMM)
4023      C *****
4024      520 CONTINUE
4025      NPTS = NPTS + II - 1
4026      II = 1
4027      NELM = JELM - 1
4028      RETURN
4029      END
4030
4031
4032
4033
4034      Subroutine CNCTT2
4035
4036      Author
4037          G.J. Smith(U. of Alberta)
4038
4039      Last Update
4040          May 1984
4041
4042      Purpose
4043          To concentrate triangular and tetrahedral
4044          elements around a pumping well.
4045
4046      Usage
4047          CALL CNCTT2(I,IM,IMM)
4048
4049      Subroutines Required
4050          None

```



```

4051 C .....
4052 C SUBROUTINE CNCTT2(I, IM, IMM)
4053 C
4054 C COMMON MEMJT(5000,32), IN(3000,8), IFACE(3500,6), YN(5000),
4055 C 1 ZN(5000)
4056 C COMMON CX(5000)
4057 C COMMON XN(5000), JMEM(5000), NBDND(4000), NDX(5000), NDY(5000)
4058 C COMMON NDZ(5000), ITYPE(5000), NDWEL(20,2,25), LEAK(100),
4059 C 1 ALEAK(100)
4060 C COMMON XIN(100), YIN(100), ISAVE(25,2), PPOSX(10), PPOSY(10),
4061 C 1 PPOSZ(10)
4062 C COMMON OBSX(10), OBSY(10), OBSZ(10), ND(16), NND(16)
4063 C COMMON COX(10), COY(10), COZ(10), IFE(6), NPMP(25), IPPN(10)
4064 C COMMON ICRNR(10,8), NFACE(6,4), ICNTST(4,2), NODE1(8)
4065 C COMMON IP, NPTS, NPIS1, NELM, IO, ITRNS, INC, IELEM, NZ, ID, YC,
4066 C 1 XC, IPN
4067 C COMMON DELX, DELY, SCL, IW1, IW2, IK, NBN, NPP, NXSTPS, NYSTPS,
4068 C 1 DELTZ
4069 C
4070 C JELM = NELM + 1
4071 C 10 FORMAT (3I5)
4072 C K = 1
4073 C II = 1
4074 C 20 FORMAT (5I5)
4075 C DO 50 J = 1, NELM
4076 C DO 40 L = 1, INC
4077 C IF (I .EQ. IN(J,L)) GO TO 30
4078 C GO TO 40
4079 C 30 ISAVE(K,1) = J
4080 C ISAVE(K,2) = L
4081 C K = K + 1
4082 C GO TO 50
4083 C 40 CONTINUE
4084 C 50 CONTINUE
4085 C KK = K - 1
4086 C TEMP = SQRT(DELX**2. + DELY**2.)
4087 C R = 0.8 * TEMP
4088 C R1 = 0.325 * TEMP
4089 C K = 1
4090 C DO 240 K1 = 1, KK, 2
4091 C IF (ID .LT. 1) CX(NPTS + II) = COX(ITRNS)
4092 C IF (ID .LT. 1) CX(NPTS + II + 1) = COX(ITRNS)
4093 C N1 = ISAVE(K1,1)
4094 C N2 = ISAVE(K1,2)
4095 C IF (ID .GT. 0) GO TO 60
4096 C GO TO (70, 110, 150), N2
4097 C IF (N2 .NE. 4) GO TO 240
4098 C IF (IN(N1,1) .NE. NDWEL(IM,2,IMM + 1)) GO TO 240
4099 C IF (ABS(XN(IN(N1,4)) - XN(IN(N1,3))) .GT. ABS(YN(IN(N1,4)) - YN(
4100 C 1 IN(N1,3))) .A ND .XN(IN(N1,3)) .GT. XN(IN(N1,4))) GO TO 70.

```



```

4101 IF (ABS(YN(IN(N1,4)) - YN(IN(N1,3))) .GT. ABS(XN(IN(N1,4)) - XN(
4102 IN(N1,3))) .A ND .YN(IN(N1,3)) .GT. YN(IN(N1,4))) GOTO110
4103 IF (ABS(XN(IN(N1,4)) - XN(IN(N1,3))) .GT. YN(IN(N1,4)) - YN(
4104 IN(N1,3))) .A ND .XN(IN(N1,3)) .LT. XN(IN(N1,4))) GOTO150
4105 IF (ABS(YN(IN(N1,4)) - YN(IN(N1,3))) .GT. ABS(XN(IN(N1,4)) - YN(
4106 IN(N1,3))) .A ND .YN(IN(N1,3)) .GT. YN(IN(N1,4))) GOTO190
4107 IF (ID .LT. 1 .AND. ISAVE(K1 - 1,2) .NE. 1) GO TO 240
4108 NM1 = 1
4109 NM2 = 2
4110 IMM1 = IMM
4111 NIN = IN(N1,2)
4112 GO TO 90
4113 IF (IMM .GT. 2) GO TO 240
4114 NM1 = 9
4115 NM2 = 10
4116 IMM1 = IMM - 1
4117 IF (ID .LT. 1) IMM1 = IMM
4118 NIN = NDZ(IN(N1,2))
4119 DO 100 NM = NM1, NM2
4120 C *****
4121 CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
4122 C *****
4123 CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, NPTS, THETA,
4124 1 NND(NM))
4125 C *****
4126 II = II + 1
4127 C *****
4128 CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
4129 C *****
4130 II = II + 1
4131 NIN = NDX(NIN)
4132 CONTINUE
4133 IF (ID .GT. 0 .AND. NM .LT. 5) GO TO 80
4134 GO TO 240
4135 IF (ID .LT. 1 .AND. ISAVE(K1 - 1,2) .NE. 2) GO TO 240
4136 NM1 = 3
4137 NM2 = 4
4138 IMM1 = IMM
4139 NIN = IN(N1,2)
4140 GO TO 130
4141 IF (IMM .GT. 2) GO TO 240
4142 NM1 = 11
4143 NM2 = 12
4144 IMM1 = IMM - 1
4145 IF (ID .LT. 1) IMM1 = IMM
4146 NIN = NDZ(IN(N1,2))
4147 DO 140 NM = NM1, NM2
4148 C *****
4149 CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
4150 C *****

```



```

4151      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
4152      *****
4153      II = II + 1
4154      *****
4155      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
4156      *****
4157      II = II + 1
4158      NIN = NDY(NIN)
4159      CONTINUE
4160      IF (ID .GT. 0 .AND. NM .LT. 6) GO TO 120
4161      GO TO 240
4162      IF (ID .LT. 1 .AND. ISAVE(K1 + 1,2) .NE. 2) GO TO 190
4163      NM1 = 5
4164      NM2 = 6
4165      IMM1 = IMM
4166      NIN = IN(N1,2)
4167      GO TO 170
4168      IF (IMM .GT. 2) GO TO 240
4169      NM1 = 13
4170      NM2 = 14
4171      NIN = NDZ(IN(N1,2))
4172      IMM1 = IMM - 1
4173      IF (ID .LT. 1) IMM1 = IMM
4174      DO 180 NM = NM1, NM2
4175      *****
4176      CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
4177      *****
4178      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, NPTS, THETA,
4179      1      NND(NM), ID)
4180      *****
4181      II = II + 1
4182      *****
4183      CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
4184      *****
4185      II = II + 1
4186      NIN = 1
4187      CONTINUE
4188      GO TO 240
4189      NM1 = 7
4190      NM2 = 8
4191      IMM1 = IMM
4192      NIN = IN(N1,2)
4193      GO TO 210
4194      IF (IMM .GT. 2) GO TO 240
4195      NM1 = 15
4196      NM2 = 16
4197      IMM1 = IMM - 1
4198      IF (ID .LT. 1) IMM1 = IMM
4199      NIN = NDZ(IN(N1,2))
4200      DO 230 NM = NM1, NM2

```



```

4201 C *****
4202     CALL INTANG(ANGLE, NDWEL(IM,2,IMM), TEMP, NIN, THETA)
4203 C *****
4204     CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R, THETA, NND(NM))
4205 C *****
4206     II = II + 1
4207 C *****
4208     CALL NDEGEN(NDWEL(IM,2,IMM1), II, ANGLE, R1, THETA, ND(NM))
4209 C *****
4210     II = II + 1
4211     DO 220 IU = 1, NPTS
4212         IF (NDY(IU) .NE. NIN) GO TO 220
4213         NIN = IU
4214         GO TO 230
4215     220 CONTINUE
4216     230 CONTINUE
4217         IF (ID .GT. 0 .AND. NM .LT. 10) GO TO 200
4218     240 CONTINUE
4219         IMM1 = IMM
4220         DO 270 NM = 1, 5, 2
4221             IN(JELM,1) = NDWEL(IM,2,IMM)
4222             IN(JELM,2) = ND(NM)
4223             IN(JELM,3) = ND(NM + 1)
4224             IF (ID .LT. 1) GO TO 250
4225             IN(JELM,4) = NND(NM + 10)
4226     250 CONTINUE
4227             JELM = JELM + 1
4228             IN(JELM,1) = NDWEL(IM,2,IMM)
4229             IN(JELM,2) = ND(NM + 1)
4230             IN(JELM,3) = ND(NM + 2)
4231             IF (ID .LT. 1) GO TO 260
4232             IN(JELM,4) = ND(NM + 10)
4233     260 CONTINUE
4234             JELM = JELM + 1
4235             IF (ID .LT. 1) GO TO 270
4236             IN(JELM,1) = ND(NM + 1)
4237             IN(JELM,2) = ND(NM + 10)
4238             IN(JELM,3) = ND(NM + 8)
4239             IN(JELM,4) = ND(NM + 9)
4240             JELM = JELM + 1
4241             IN(JELM,1) = NDWEL(IM,2,IMM)
4242             IN(JELM,2) = ND(NM + 8)
4243             IN(JELM,3) = ND(NM + 10)
4244             IN(JELM,4) = NDWEL(IM,2,IMM1)
4245             JELM = JELM + 1
4246             IN(JELM,1) = NDWEL(IM,2,IMM)
4247             IN(JELM,2) = ND(NM + 1)
4248             IN(JELM,3) = ND(NM + 10)
4249             IN(JELM,4) = ND(NM + 8)
4250             JELM = JELM + 1

```



```

4251      270 CONTINUE
4252      IN(JELM,1) = ND(1)
4253      IN(JELM,2) = NDWEL(IM,2,IMM)
4254      IN(JELM,3) = ND(8)
4255      IF (ID.LT.1) GO TO 280
4256      IN(JELM,4) = ND(9)
4257
4258      280 CONTINUE
4259      JELM = JELM + 1
4260      IN(JELM,1) = ND(8)
4261      IN(JELM,2) = NDWEL(IM,2,IMM)
4262      IN(JELM,3) = ND(7)
4263      IF (ID.LT.1) GO TO 290
4264      IN(JELM,4) = ND(16)
4265
4266      290 CONTINUE
4267      JELM = JELM + 1
4268      DO 350 NM = 1, 5, 2
4269          IN(JELM,1) = ND(NM)
4270          IN(JELM,2) = NND(NM)
4271          IN(JELM,3) = ND(NM + 1)
4272          IF (ID.LT.1) GO TO 300
4273          IN(JELM,4) = ND(NM + 8)
4274
4275      300 CONTINUE
4276      JELM = JELM + 1
4277      IN(JELM,1) = NND(NM + 1)
4278      IN(JELM,2) = ND(NM + 1)
4279      IN(JELM,3) = NND(NM)
4280      IF (ID.LT.1) GO TO 310
4281      IN(JELM,4) = NND(NM + 9)
4282
4283      310 CONTINUE
4284      JELM = JELM + 1
4285      IF (ID.LT.1) GO TO 320
4286      IN(JELM,1) = NND(NM)
4287      IN(JELM,2) = NND(NM + 9)
4288      IN(JELM,3) = ND(NM + 8)
4289      IN(JELM,4) = NND(NM + 8)
4290
4291      JELM = JELM + 1
4292      IN(JELM,1) = ND(NM + 1)
4293      IN(JELM,2) = ND(NM + 8)
4294      IN(JELM,3) = NND(NM + 9)
4295      IN(JELM,4) = ND(NM + 9)
4296
4297      JELM = JELM + 1
4298      IN(JELM,1) = ND(NM + 1)
4299      IN(JELM,2) = NND(NM)
4300      IN(JELM,3) = NND(NM + 9)
4301      IN(JELM,4) = ND(NM + 8)
4302
4303      JELM = JELM + 1
4304      CONTINUE
4305      IN(JELM,1) = ND(NM + 2)
4306      IN(JELM,2) = ND(NM + 1)
4307      IN(JELM,3) = NND(NM + 2)

```



```

4301 IF (ID .LT. 1) GO TO 330
4302 IN(JELM,4) = ND(NM + 10)
4303 CONTINUE
4304 JELM = JELM + 1
4305 IN(JELM,1) = NND(NM + 1)
4306 IN(JELM,2) = NND(NM + 2)
4307 IN(JELM,3) = ND(NM + 2)
4308 IF (ID .LT. 1) GO TO 340
4309 IN(JELM,4) = NND(NM + 9)
4310 CONTINUE
4311 JELM = JELM + 1
4312 IF (ID .LT. 1) GO TO 350
4313 IN(JELM,1) = ND(NM + 1)
4314 IN(JELM,2) = NND(NM + 9)
4315 IN(JELM,3) = ND(NM + 10)
4316 IN(JELM,4) = ND(NM + 9)
4317 JELM = JELM + 1
4318 IN(JELM,1) = NND(NM + 2)
4319 IN(JELM,2) = ND(NM + 10)
4320 IN(JELM,3) = NND(NM + 9)
4321 IN(JELM,4) = NND(NM + 10)
4322 JELM = JELM + 1
4323 IN(JELM,1) = ND(NM + 1)
4324 IN(JELM,2) = ND(NM + 2)
4325 IN(JELM,3) = ND(NM + 10)
4326 IN(JELM,4) = NND(NM + 9)
4327 JELM = JELM + 1
4328 CONTINUE
4329 IN(JELM,1) = ND(7)
4330 IN(JELM,2) = NND(7)
4331 IN(JELM,3) = ND(8)
4332 IF (ID .LT. 1) GO TO 360
4333 IN(JELM,4) = ND(15)
4334 CONTINUE
4335 JELM = JELM + 1
4336 IN(JELM,1) = NND(8)
4337 IN(JELM,2) = ND(8)
4338 IN(JELM,3) = NND(7)
4339 IF (ID .LT. 1) GO TO 370
4340 IN(JELM,4) = NND(16)
4341 CONTINUE
4342 JELM = JELM + 1
4343 IN(JELM,1) = ND(1)
4344 IN(JELM,2) = ND(8)
4345 IN(JELM,3) = NND(1)
4346 IF (ID .LT. 1) GO TO 380
4347 IN(JELM,4) = ND(9)
4348 CONTINUE
4349 JELM = JELM + 1
4350 IN(JELM,1) = NND(8)

```



```

4351 IN(JELM,2) = NND(1)
4352 IN(JELM,3) = ND(8)
4353 IF (ID.LT. 1) GO TO 390
4354 IN(JELM,4) = NND(16)
4355
4356 390 CONTINUE
4357 JELM = JELM + 1
4358 DO 490 K1 = 1, KK, 2
4359   N1 = ISAVE(K1,1)
4360   N2 = ISAVE(K1,2)
4361   IF (ID.LT. 1) GO TO 400
4362   GO TO (410, 430, 450), N2
4363   IF (N2.NE. 4) GO TO 240
4364   IF (IN(N1,1).NE. NDWEL(IM,2,IMM + 1)) GO TO 240
4365   IF (ABS(XN(IN(N1,4)) - XN(IN(N1,3))) .GT. ABS(YN(IN(N1,4)) - YN(
1   IN(N1,3))) .A ND .XN(IN(N1,3)) .GT. XN(IN(N1,4))) GOTO410
4366   IF (ABS(YN(IN(N1,4)) - YN(IN(N1,3))) .GT. ABS(XN(IN(N1,4)) - XN(
1   IN(N1,3))) .A ND .YN(IN(N1,3)) .GT. YN(IN(N1,4))) GOTO430
4367   IF (ABS(XN(IN(N1,4)) - XN(IN(N1,3))) .GT. ABS(YN(IN(N1,4)) - YN(
1   IN(N1,3))) .A ND .XN(IN(N1,3)) .LT. XN(IN(N1,4))) GOTO450
4368   IF (ABS(YN(IN(N1,4)) - YN(IN(N1,3))) .GT. ABS(XN(IN(N1,4)) - YN(
1   IN(N1,3))) .A ND .YN(IN(N1,3)) .GT. YN(IN(N1,4))) GOTO470
4369   IF (ID.GT. 0) GO TO 420
4370   IF (ISAVE(K1 - 1,2).NE. 1) GO TO 490
4371   IN(JELM,1) = NND(1)
4372   IN(JELM,2) = IN(N1,3)
4373   IN(JELM,3) = NND(2)
4374   JELM = JELM + 1
4375   IN(JELM,1) = NND(2)
4376   IN(JELM,2) = IN(N1,3)
4377   IN(JELM,3) = NND(3)
4378   JELM = JELM + 1
4379   IN(N1,1) = NND(1)
4380   IN(N1 + 1,1) = NND(3)
4381   GO TO 490
4382   IN(JELM,1) = IN(N1 - 3,1)
4383   IN(JELM,2) = IN(N1 - 3,2)
4384   IN(JELM,3) = NND(1)
4385   IN(JELM,4) = IN(N1 - 3,4)
4386   JELM = JELM + 1
4387   IN(JELM,1) = IN(N1 - 2,1)
4388   IN(JELM,2) = NND(10)
4389   IN(JELM,3) = IN(N1 - 2,3)
4390   IN(JELM,4) = IN(N1 - 2,4)
4391   JELM = JELM + 1
4392   IN(JELM,1) = NND(2)
4393   IN(JELM,2) = NND(1)
4394   IN(JELM,3) = IN(N1 - 1,3)
4395   IN(JELM,4) = NND(10)
4396   JELM = JELM + 1
4397   IN(JELM,1) = NND(1)
4398   IN(JELM,2) = NND(2)
4399   IN(JELM,3) = IN(N1 - 1,3)
4400   IN(JELM,4) = NND(10)

```



```

4401      IN(JELM,2) = IN(N1,2)
4402      IN(JELM,3) = NND(10)
4403      IN(JELM,4) = NND(9)
4404      JELM = JELM + 1
4405      IN(JELM,1) = NND(1)
4406      IN(JELM,2) = IN(N1 + 1,2)
4407      IN(JELM,3) = NND(10)
4408      IN(JELM,4) = IN(N1 + 1,4)
4409      JELM = JELM + 1
4410      IN(N1 - 3,1) = NND(2)
4411      IN(N1 - 3,3) = NND(3)
4412      IN(N1 - 3,4) = NND(10)
4413      IN(N1 - 2,3) = NND(10)
4414      IN(N1 - 1,2) = NND(3)
4415      IN(N1,1) = NND(3)
4416      IN(N1,2) = NND(10)
4417      IN(N1,4) = NND(11)
4418      IN(N1 + 1,1) = NND(3)
4419      IN(N1 + 1,4) = NND(10)
4420      GO TO 490
4421      IF (ID .GT. 0) GO TO 440
4422      IF (ISAVE(K1 - 1,2) .NE. 2) GO TO 490
4423      IN(JELM,1) = IN(N1,3)
4424      IN(JELM,2) = NND(5)
4425      IN(JELM,3) = NND(4)
4426      JELM = JELM + 1
4427      IN(JELM,1) = IN(N1,3)
4428      IN(JELM,2) = NND(4)
4429      IN(JELM,3) = NND(3)
4430      JELM = JELM + 1
4431      IN(N1,2) = NND(5)
4432      IN(N1 + 1,2) = NND(3)
4433      GO TO 490
4434      IN(JELM,1) = NND(4)
4435      IN(JELM,2) = NND(3)
4436      IN(JELM,3) = IN(N1 - 1,3)
4437      IN(JELM,4) = NND(12)
4438      JELM = JELM + 1
4439      IN(JELM,1) = NND(3)
4440      IN(JELM,2) = IN(N1,2)
4441      IN(JELM,3) = NND(12)
4442      IN(JELM,4) = NND(11)
4443      JELM = JELM + 1
4444      IN(JELM,1) = IN(N1 + 1,1)
4445      IN(JELM,2) = IN(N1 + 1,2)
4446      IN(JELM,3) = NND(3)
4447      IN(JELM,4) = IN(N1 + 1,4)
4448      JELM = JELM + 1
4449      IN(JELM,1) = IN(N1 + 2,1)
4450      IN(JELM,2) = NND(12)

```



```

4451 IN(JELM,3) = IN(N1 + 2,3)
4452 IN(JELM,4) = IN(N1 + 2,4)
4453 JELM = JELM + 1
4454 IN(JELM,1) = IN(N1 + 3,1)
4455 IN(JELM,2) = NND(3)
4456 IN(JELM,3) = IN(N1 + 3,3)
4457 IN(JELM,4) = NND(12)
4458 JELM = JELM + 1
4459 IN(N1 - 1,2) = NND(5)
4460 IN(N1,1) = NND(5)
4461 IN(N1,2) = NND(12)
4462 IN(N1,4) = NND(13)
4463 IN(N1 + 1,1) = NND(4)
4464 IN(N1 + 1,3) = NND(5)
4465 IN(N1 + 1,4) = NND(12)
4466 IN(N1 + 2,3) = NND(12)
4467 IN(N1 + 3,2) = NND(5)
4468 IN(N1 + 3,3) = NND(12)
4469 GO TO 490
4470 IF (ID .GT. 0) GO TO 460
4471 IF (ISAVE(K1 + 1,2) .NE. 2) GO TO 470
4472 IN(JELM,1) = IN(N1,1)
4473 IN(JELM,2) = NND(6)
4474 IN(JELM,3) = NND(5)
4475 JELM = JELM + 1
4476 IN(JELM,1) = IN(N1,1)
4477 IN(JELM,2) = NND(7)
4478 IN(JELM,3) = NND(6)
4479 JELM = JELM + 1
4480 IN(N1,3) = NND(7)
4481 IN(N1 + 1,2) = NND(5)
4482 GO TO 490
4483 IN(JELM,1) = NND(6)
4484 IN(JELM,2) = NND(5)
4485 IN(JELM,3) = IN(N1 - 1,3)
4486 IN(JELM,4) = NND(14)
4487 JELM = JELM + 1
4488 IN(JELM,1) = NND(5)
4489 IN(JELM,2) = IN(N1,2)
4490 IN(JELM,3) = NND(14)
4491 IN(JELM,4) = NND(13)
4492 JELM = JELM + 1
4493 IN(JELM,1) = IN(N1 + 1,1)
4494 IN(JELM,2) = IN(N1 + 1,2)
4495 IN(JELM,3) = NND(5)
4496 IN(JELM,4) = IN(N1 + 1,4)
4497 JELM = JELM + 1
4498 IN(JELM,1) = IN(N1 + 2,1)
4499 IN(JELM,2) = NND(14)
4500 IN(JELM,3) = IN(N1 + 2,3)

```



```

4501 IN(JELM,4) = IN(N1 + 2,4)
4502 JELM = JELM + 1
4503 IN(JELM,1) = IN(N1 + 3,1)
4504 IN(JELM,2) = NND(5)
4505 IN(JELM,3) = IN(N1 + 3,3)
4506 IN(JELM,4) = NND(14)
4507 JELM = JELM + 1
4508 IN(N1 - 1,2) = NND(7)
4509 IN(N1,1) = NND(7)
4510 IN(N1,2) = NND(14)
4511 IN(N1,4) = NND(15)
4512 IN(N1 + 1,1) = NND(6)
4513 IN(N1 + 1,3) = NND(7)
4514 IN(N1 + 1,4) = NND(14)
4515 IN(N1 + 2,3) = NND(14)
4516 IN(N1 + 3,2) = NND(7)
4517 IN(N1 + 3,3) = NND(14)
4518 GO TO 490
4519 IF (ID.GT. O) GO TO 480
4520 IN(JELM,1) = NND(7)
4521 IN(JELM,2) = IN(N1,2)
4522 IN(JELM,3) = NND(8)
4523 JELM = JELM + 1
4524 IN(JELM,1) = NND(8)
4525 IN(JELM,2) = IN(N1,2)
4526 IN(JELM,3) = NND(1)
4527 JELM = JELM + 1
4528 IN(N1,3) = NND(7)
4529 IN(N1 + 1,1) = NND(1)
4530 GO TO 490
4531 IN(JELM,1) = IN(N1 - 3,1)
4532 IN(JELM,2) = IN(N1 - 3,2)
4533 IN(JELM,3) = NND(7)
4534 IN(JELM,4) = IN(N1 - 3,4)
4535 JELM = JELM + 1
4536 IN(JELM,1) = IN(N1 - 2,1)
4537 IN(JELM,2) = NND(16)
4538 IN(JELM,3) = IN(N1 - 2,3)
4539 IN(JELM,4) = IN(N1 - 2,4)
4540 JELM = JELM + 1
4541 IN(JELM,1) = NND(8)
4542 IN(JELM,2) = NND(7)
4543 IN(JELM,3) = IN(N1 - 1,3)
4544 IN(JELM,4) = NND(16)
4545 JELM = JELM + 1
4546 IN(JELM,1) = NND(7)
4547 IN(JELM,2) = IN(N1,2)
4548 IN(JELM,3) = NND(16)
4549 IN(JELM,4) = NND(15)
4550 JELM = JELM + 1

```

470

480


```

4551 IN(JELM,1) = NND(7)
4552 IN(JELM,2) = IN(N1 + 1,2)
4553 IN(JELM,3) = NND(16)
4554 IN(JELM,4) = IN(N1 + 1,4)
4555 JELM = JELM + 1
4556 IN(N1 - 3,1) = NND(8)
4557 IN(N1 - 3,3) = NND(1)
4558 IN(N1 - 3,4) = NND(16)
4559 IN(N1 - 2,3) = NND(16)
4560 IN(N1 - 1,2) = NND(1)
4561 IN(N1,1) = NND(1)
4562 IN(N1,2) = NND(16)
4563 IN(N1,4) = NND(9)
4564 IN(N1 + 1,1) = NND(1)
4565 IN(N1 + 1,4) = NND(16)
4566
4567 490 CONTINUE
4568 IF (ID .LT. 1) GO TO 510
4569 DO 500 I = 1, 8
4570 ND(I + 8) = ND(I)
4571 NND(I + 8) = NND(I)
4572 500 CONTINUE
4573 510 CONTINUE
4574 NPTS = NPTS + II - 1
4575 II = 1
4576 NELM = JELM - 1
4577 RETURN
4578 END

```


APPENDIX C: Results of a Two-Dimensional Simulation

40	7227.63	28840.59	0.0258
41	10863.45	27494.82	0.0258
42	14175.17	22360.49	0.0129
43	17485.53	21677.71	0.0129
44	20867.76	22258.47	0.0258
45	4315.79	32025.07	0.0129
46	26324.20	13159.38	0.0258
47	26042.10	10876.24	0.0258
48	26616.01	15610.16	0.0258
49	26950.79	18271.25	0.0258
50	27262.17	20943.54	0.0258
51	23973.27	22829.91	0.0258
52	27331.37	23213.26	0.0258
53	8005.41	32244.25	0.0129
54	11695.02	30600.95	0.0129
55	14499.27	24900.70	0.0129
56	18135.10	23902.39	0.0129
57	21305.45	24271.19	0.0258
58	24311.75	24752.32	0.0258
59	7617.56	35139.63	0.0258
60	10889.89	35647.92	0.0129
61	28070.85	10162.00	0.0258
62	28761.99	12789.68	0.0258
63	29482.92	15515.75	0.0258
64	30267.54	18301.15	0.0258
65	31462.40	21131.85	0.0258
66	30740.48	23590.60	0.0258
67	27292.56	25374.36	0.0258
68	30249.75	26193.65	0.0258
69	14054.93	33707.08	0.0129
70	15384.64	27440.91	0.0129
71	17219.98	29981.12	0.0129
72	19074.26	26127.07	0.0129
73	21912.01	26343.88	0.0258
74	24588.02	26816.16	0.0258
75	27241.14	27646.80	0.0258
76	31167.14	12323.90	0.0258
77	30099.60	9483.86	0.0258
78	32234.43	15309.57	0.0258
79	33360.42	18481.51	0.0258
80	33318.84	21169.92	0.0258
81	33898.13	23975.45	0.0258
82	33165.87	27061.98	0.0258
83	29878.95	28746.97	0.0258
84	32503.95	29941.33	0.0258
85	20385.03	28351.76	0.0129
86	22628.35	28149.02	0.0258
87	24871.68	28712.74	0.0258
88	27115.00	29711.13	0.0258
89	29358.33	31252.10	0.0258

90	32128.34	8506.86	0.0258
91	33569.63	11796.49	0.0258
92	34890.98	15022.41	0.0258
93	35798.70	17805.59	0.0258
94	34819.30	21338.53	0.0258
95	36384.81	19533.96	0.0258
96	34969.76	19929.60	0.0258
97	36841.91	23401.88	0.0258
98	36500.46	25069.06	0.0258
99	36074.18	27889.37	0.0258
100	35305.93	22839.99	0.0258
101	35125.77	31015.34	0.0258
102	31601.66	33004.20	0.0258
103	33844.99	34307.98	0.0258
104	35952.28	11352.43	0.0258
105	34157.09	7693.09	0.0258
106	37591.07	14856.54	0.0258
107	38669.75	18484.30	0.0258
108	36023.16	22163.21	0.0258
109	35801.34	21380.29	0.0258
110	35874.18	20668.94	0.0258
111	36618.36	20502.21	0.0258
112	37319.96	20592.63	0.0258
113	37799.95	19859.51	0.0258
114	39385.82	24731.88	0.0258
115	37425.82	22065.20	0.0258
116	36802.19	22324.44	0.0258
117	38400.97	23122.17	0.0258
118	39034.49	28494.37	0.0258
119	37748.10	31879.06	0.0258
120	36088.32	35434.13	0.0258
121	36185.84	7240.91	0.0258
122	38433.77	11061.90	0.0258
123	40363.65	14705.58	0.0258
124	41779.31	18219.23	0.0258
125	38558.93	21478.49	0.0258
126	39882.98	21531.49	0.0258
127	42549.38	21749.37	0.0258
128	36688.13	21388.36	0.0258
129	37640.23	21410.09	0.0258
130	42701.86	25388.77	0.0258
131	41885.67	28927.87	0.0258
132	40371.48	32521.27	0.0258
133	38331.65	36276.93	0.0258
134	40899.41	11035.02	0.0258
135	38214.58	7146.67	0.0258
136	43277.66	14707.91	0.0258
137	44820.54	18345.17	0.0258
138	45766.28	21965.98	0.0258
139	45749.89	25584.13	0.0258

Q

140	44911.98	29232.55	0.0258
141	43021.42	32921.56	0.0258
142	40574.97	36782.93	0.0258
143	40243.33	7362.59	0.0258
144	43720.77	11231.36	0.0258
145	46315.02	14920.78	0.0258
146	48154.61	18538.38	0.0258
147	49129.88	22143.05	0.0258
148	49269.19	25757.88	0.0258
149	48111.85	29365.49	0.0258
150	45905.13	33062.23	0.0258
151	42818.30	36966.16	0.0258
152	47059.22	11692.51	0.0258
153	42272.10	8153.72	0.0258
154	49874.09	15231.31	0.0258
155	51840.77	18770.10	0.0258
156	53092.34	22308.89	0.0258
157	53370.24	25847.68	0.0258
158	52133.61	29386.48	0.0258
159	49382.28	32925.27	0.0258
160	45061.63	36464.09	0.0258

ELEMENT DATA ELEMENT	INCIDENCE			LEAKANCE
	1	2	3	
1	17	18	25	0.0
2	25	18	26	0.0
3	25	26	34	0.0
4	25	34	35	0.0
5	35	34	46	0.0
6	46	34	47	0.0
7	46	47	61	0.0
8	46	61	62	0.0
9	62	61	76	0.0
10	76	61	77	0.0
11	76	77	90	0.0
12	76	90	91	0.0
13	91	90	104	0.0
14	104	90	105	0.0
15	104	105	121	0.0
16	104	121	122	0.0
17	122	121	134	0.0
18	134	121	135	0.0
19	134	135	143	0.0
20	134	143	144	0.0
21	144	143	152	0.0
22	152	143	153	0.0
23	19	17	25	0.0

24	19	25	27	0.0
25	27	25	36	0.0
26	36	25	35	0.0
27	36	35	46	0.0
28	36	46	48	0.0
29	48	46	63	0.0
30	63	46	62	0.0
31	63	62	76	0.0
32	63	76	78	0.0
33	78	76	92	0.0
34	92	76	91	0.0
35	92	91	104	0.0
36	92	104	106	0.0
37	106	104	123	0.0
38	123	104	122	0.0
39	123	122	134	0.0
40	123	134	136	0.0
41	136	134	145	0.0
42	145	134	144	0.0
43	145	144	152	0.0
44	145	152	154	0.0
45	23	19	28	0.0
46	28	19	27	0.0
47	28	27	36	0.0
48	28	36	37	0.0
49	37	36	49	0.0
50	49	36	48	0.0
51	49	48	63	0.0
52	49	63	64	0.0
53	64	63	79	0.0
54	79	63	78	0.0
55	79	78	92	0.0
56	79	92	93	0.0
57	93	92	107	0.0
58	107	92	106	0.0
59	107	106	123	0.0
60	107	123	124	0.0
61	124	123	137	0.0
62	137	123	136	0.0
63	137	136	145	0.0
64	137	145	146	0.0
65	146	145	155	0.0
66	155	145	154	0.0
67	32	23	28	0.0
68	32	28	38	0.0
69	38	28	39	0.0
70	39	28	37	0.0
71	39	37	49	0.0
72	39	49	50	0.0
73	50	49	65	0.0

74	65	49	64	0.0
75	65	64	79	0.0
76	65	79	80	0.0
77	80	79	94	0.0
78	95	79	93	0.0
79	95	93	107	0.0
80	125	107	126	0.0
81	126	107	127	0.0
82	127	107	124	0.0
83	127	124	137	0.0
84	127	137	138	0.0
85	138	137	147	0.0
86	147	137	146	0.0
87	147	146	155	0.0
88	147	155	156	0.0
89	43	32	44	0.0
90	44	32	38	0.0
91	44	38	39	0.0
92	44	39	51	0.0
93	51	39	52	0.0
94	52	39	50	0.0
95	52	50	65	0.0
96	52	65	66	0.0
97	66	65	81	0.0
98	81	65	80	0.0
99	81	80	94	0.0
100	81	97	98	0.0
101	98	97	114	0.0
102	114	125	126	0.0
103	114	126	127	0.0
104	114	127	130	0.0
105	130	127	139	0.0
106	139	127	138	0.0
107	139	138	147	0.0
108	139	147	148	0.0
109	148	147	157	0.0
110	157	147	156	0.0
111	56	43	44	0.0
112	56	44	57	0.0
113	57	44	58	0.0
114	58	44	51	0.0
115	58	51	52	0.0
116	58	52	67	0.0
117	67	52	68	0.0
118	68	52	66	0.0
119	68	66	81	0.0
120	68	81	82	0.0
121	82	81	99	0.0
122	99	81	98	0.0
123	99	98	114	0.0

124	99	114	118	0.0
125	118	114	131	0.0
126	131	114	130	0.0
127	131	130	139	0.0
128	131	139	140	0.0
129	140	139	149	0.0
130	149	139	148	0.0
131	149	148	157	0.0
132	149	157	158	0.0
133	72	56	73	0.0
134	73	56	57	0.0
135	73	57	58	0.0
136	73	58	74	0.0
137	74	58	75	0.0
138	75	58	67	0.0
139	75	67	68	0.0
140	75	68	83	0.0
141	83	68	84	0.0
142	84	68	82	0.0
143	84	82	99	0.0
144	84	99	101	0.0
145	101	99	119	0.0
146	119	99	118	0.0
147	119	118	131	0.0
148	119	131	132	0.0
149	132	131	141	0.0
150	141	131	140	0.0
151	141	140	149	0.0
152	141	149	150	0.0
153	150	149	159	0.0
154	159	149	158	0.0
155	85	72	73	0.0
156	85	73	86	0.0
157	86	73	87	0.0
158	87	73	74	0.0
159	87	74	75	0.0
160	87	75	88	0.0
161	88	75	89	0.0
162	89	75	83	0.0
163	89	83	84	0.0
164	89	84	102	0.0
165	102	84	103	0.0
166	103	84	101	0.0
167	103	101	119	0.0
168	103	119	120	0.0
169	120	119	133	0.0
170	133	119	132	0.0
171	133	132	141	0.0
172	133	141	142	0.0
173	142	141	151	0.0

174	151	141	150	0.0
175	151	150	159	0.0
176	151	159	160	0.0
177	128	111	112	0.0
178	128	112	129	0.0
179	128	129	115	0.0
180	128	115	116	0.0
181	128	116	108	0.0
182	128	108	109	0.0
183	128	109	110	0.0
184	128	110	111	0.0
185	111	95	112	0.0
186	112	95	113	0.0
187	112	125	129	0.0
188	112	113	125	0.0
189	129	125	115	0.0
190	115	125	117	0.0
191	115	97	116	0.0
192	115	117	97	0.0
193	116	97	108	0.0
194	108	97	100	0.0
195	108	94	109	0.0
196	108	100	94	0.0
197	109	94	110	0.0
198	94	96	110	0.0
199	110	95	111	0.0
200	110	96	95	0.0
201	94	79	96	0.0
202	96	79	95	0.0
203	95	107	113	0.0
204	113	107	125	0.0
205	81	100	97	0.0
206	81	94	100	0.0
207	114	97	117	0.0
208	114	117	125	0.0
209	2	1	3	0.0
210	2	3	4	0.0
211	4	3	6	0.0
212	6	3	7	0.0
213	6	7	11	0.0
214	6	11	12	0.0
215	12	11	17	1.OE-08
216	17	11	18	1.OE-08
217	5	2	4	0.0
218	5	4	8	0.0
219	8	4	9	0.0
220	9	4	6	0.0
221	9	6	12	0.0
222	9	12	15	0.0
223	15	12	19	1.OE-08

224	19	12	17	1.0E-08
225	10	5	8	0.0
226	10	8	13	0.0
227	13	8	14	0.0
228	14	8	9	0.0
229	14	9	15	0.0
230	14	15	22	0.0
231	22	15	23	1.0E-08
232	23	15	19	1.0E-08
233	16	10	13	0.0
234	16	13	20	0.0
235	20	13	21	0.0
236	21	13	14	0.0
237	21	14	22	0.0
238	21	22	31	0.0
239	31	22	32	1.0E-08
240	32	22	23	1.0E-08
241	24	16	20	0.0
242	24	20	29	0.0
243	29	20	30	0.0
244	30	20	21	0.0
245	30	21	31	0.0
246	30	31	42	0.0
247	42	31	43	1.0E-08
248	43	31	32	1.0E-08
249	33	24	29	0.0
250	33	29	40	0.0
251	40	29	41	0.0
252	41	29	30	0.0
253	41	30	42	0.0
254	41	42	55	0.0
255	55	42	56	1.0E-08
256	56	42	43	1.0E-08
257	45	33	40	0.0
258	45	40	53	0.0
259	53	40	54	0.0
260	54	40	41	0.0
261	54	41	55	0.0
262	54	55	70	0.0
263	70	55	72	1.0E-08
264	72	55	56	1.0E-08
265	59	45	60	0.0
266	60	45	53	0.0
267	60	53	69	0.0
268	69	53	54	0.0
269	69	54	71	0.0
270	71	54	70	0.0
271	71	70	72	1.0E-08
272	71	72	85	1.0E-08

STORAGE COEFFICIENT 8.0E-04

NUMBER OF CONSTANT-HEAD NODES 0
NUMBER OF DEGREES OF FREEDOM 160

INTERVAL FROM 0.0 TO 0.0014 DAYS
WELL PW 1 NODE 1 DISCHARGE 1.3380

PIEZOMETRIC SURFACE (NODE AND DRAWDOWN)

TIME		0.0003 DAYS		30. SECONDS											
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00000	26	-0.00000	27	-0.00000	28	-0.00000	29	-0.00000	30	0.00000	31	-0.00000	32	0.00000
33	0.00000	34	0.00000	35	0.00000	36	0.00000	37	0.00000	38	-0.00000	39	0.00001	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00000	45	0.00000	46	-0.00001	47	0.00000	48	-0.00000
49	-0.00002	50	-0.00001	51	0.00000	52	-0.00002	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00000	58	0.00000	59	-0.00000	60	-0.00000	61	0.00001	62	-0.00000	63	0.00003	64	0.00004
65	0.00008	66	0.00004	67	-0.00000	68	0.00003	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00000	74	0.00000	75	-0.00001	76	-0.00002	77	0.00001	78	0.00004	79	-0.00026	80	-0.00040
81	-0.00025	82	0.00004	83	-0.00000	84	-0.00002	85	0.00000	86	-0.00000	87	0.00000	88	0.00000
89	0.00001	90	0.00001	91	-0.00001	92	0.00009	93	-0.00032	94	0.00229	95	0.00215	96	0.00071
97	0.00211	98	-0.00030	99	0.00009	100	0.00074	101	-0.00001	102	0.00001	103	0.00001	104	-0.00002
105	0.00000	106	0.00003	107	-0.00022	108	-0.01006	109	-0.02595	110	-0.00996	111	-0.02572	112	-0.01101
113	0.00085	114	-0.00021	115	-0.00869	116	-0.02571	117	0.00087	118	0.00003	119	-0.00002	120	0.00000
121	0.00001	122	-0.00000	123	0.00003	124	0.00003	125	0.00232	126	-0.00030	127	0.00008	128	0.13083
129	-0.02822	130	0.00003	131	0.00002	132	-0.00000	133	0.00001	134	-0.00000	135	0.00000	136	-0.00000
137	-0.00002	138	-0.00001	139	-0.00002	140	-0.00000	141	-0.00000	142	0.00000	143	0.00000	144	0.00000
145	0.00000	146	0.00000	147	0.00001	148	0.00000	149	0.00000	150	0.00000	151	0.00000	152	-0.00000
153	-0.00000	154	-0.00000	155	-0.00000	156	-0.00000	157	-0.00000	158	-0.00000	159	-0.00000	160	-0.00000

41	0.00000	42	-0.00000	43	0.00000	44	-0.00000	45	0.00000	46	-0.00001	47	0.00000	48	-0.00000
49	-0.00004	50	-0.00002	51	0.00001	52	-0.00004	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00000	58	0.00001	59	-0.00000	60	-0.00000	61	0.00001	62	-0.00001	63	0.00006	64	0.00008
65	0.00016	66	0.00008	67	-0.00000	68	0.00006	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00000	74	0.00000	75	-0.00001	76	-0.00004	77	0.00001	78	0.00008	79	-0.00051	80	-0.00078
81	-0.00050	82	0.00008	83	-0.00001	84	-0.00004	85	0.00000	86	-0.00000	87	0.00000	88	0.00000
89	0.00001	90	0.00003	91	-0.00003	92	0.00018	93	-0.00063	94	0.00450	95	0.00424	96	0.00146
97	0.00415	98	-0.00060	99	0.00017	100	0.00150	101	-0.00003	102	0.00001	103	0.00002	104	-0.00004
105	0.00001	106	0.00006	107	-0.00043	108	-0.02013	109	-0.05075	110	-0.01999	111	-0.05030	112	-0.02202
113	0.00171	114	-0.00041	115	-0.01742	116	-0.05039	117	0.00178	118	0.00006	119	-0.00004	120	0.00001
121	0.00001	122	-0.00000	123	0.00005	124	0.00007	125	0.00456	126	-0.00059	127	0.00015	128	0.25999
129	-0.05529	130	0.00006	131	0.00005	132	-0.00000	133	0.00001	134	-0.00001	135	0.00000	136	-0.00000
137	-0.00003	138	-0.00002	139	-0.00003	140	-0.00000	141	-0.00001	142	0.00000	143	0.00000	144	0.00000
145	0.00001	146	0.00000	147	0.00001	148	0.00000	149	0.00001	150	0.00000	151	0.00000	152	-0.00000
153	-0.00000	154	-0.00000	155	-0.00000	156	-0.00000	157	-0.00000	158	-0.00000	159	-0.00000	160	-0.00000

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88
89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120
121	122	123	124	125	126	127	128
129	130	131	132	133	134	135	136
137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152
153	154	155	156	157	158	159	160

TIME		0.0014 DAYS		121. SECONDS													
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000		
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000		
17	0.00000	18	0.00000	19	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000		
25	-0.00000	26	-0.00000	27	-0.00000	28	-0.00001	29	-0.00000	30	0.00000	31	-0.00000	32	0.00000		
33	0.00000	34	0.00001	35	0.00000	36	0.00001	37	0.00001	38	-0.00000	39	0.00003	40	-0.00000		
41	0.00000	42	-0.00000	43	0.00000	44	-0.00001	45	0.00000	46	-0.00002	47	0.00000	48	-0.00001		
49	-0.00007	50	-0.00005	51	0.00001	52	-0.00008	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000		

363. SECONDS

0.0042 DAYS

TIME

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00000	28	-0.00002	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00001	35	0.00000	36	0.00004	37	0.00002	38	-0.00001	39	0.00007	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00001	45	0.00000	46	-0.00005	47	0.00001	48	-0.00003
49	-0.00019	50	-0.00012	51	0.00003	52	-0.00021	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00000	58	0.00004	59	-0.00000	60	-0.00000	61	0.00007	62	-0.00003	63	0.00031	64	0.00041
65	0.00076	66	0.00040	67	-0.00002	68	0.00031	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00000	75	-0.00005	76	-0.00023	77	0.00006	78	0.00044	79	-0.00267	80	-0.00350
81	-0.00255	82	0.00042	83	-0.00003	84	-0.00021	85	0.00000	86	-0.00000	87	0.00001	88	0.00001
89	0.00007	90	0.00013	91	-0.00014	92	0.00086	93	-0.00288	94	0.02186	95	0.02074	96	0.01052
97	0.02048	98	-0.00277	99	0.00082	100	0.00938	101	-0.00013	102	0.00006	103	0.00012	104	-0.00019
105	0.00005	106	0.00031	107	-0.00216	108	-0.11824	109	-0.23422	110	-0.12167	111	-0.23219	112	-0.12867
113	0.00991	114	-0.00212	115	-0.10401	116	-0.23920	117	0.01225	118	0.00031	119	-0.00018	120	0.00005
121	0.00006	122	-0.00002	123	0.00025	124	0.00035	125	0.02183	126	-0.00257	127	0.00071	128	1.45062
129	-0.26182	130	0.00035	131	0.00024	132	-0.00002	133	0.00005	134	-0.00004	135	0.00000	136	-0.00002
137	-0.00015	138	-0.00011	139	-0.00015	140	-0.00002	141	-0.00004	142	0.00000	143	0.00001	144	0.00000
145	0.00003	146	0.00002	147	0.00005	148	0.00002	149	0.00003	150	0.00000	151	0.00001	152	-0.00001
153	0.00000	154	-0.00001	155	-0.00002	156	-0.00001	157	-0.00002	158	-0.00001	159	-0.00001	160	-0.00000

484. SECONDS

0.0056 DAYS

TIME

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00000	28	-0.00002	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00002	35	0.00000	36	0.00004	37	0.00003	38	-0.00001	39	0.00009	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00002	45	0.00000	46	-0.00007	47	0.00001	48	-0.00003
49	-0.00024	50	-0.00014	51	0.00004	52	-0.00026	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00001	58	0.00004	59	-0.00000	60	-0.00000	61	0.00009	62	-0.00003	63	0.00040	64	0.00052
65	0.00094	66	0.00050	67	-0.00003	68	0.00039	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00000	75	-0.00007	76	-0.00028	77	0.00008	78	0.00056	79	-0.00339	80	-0.00419
81	-0.00320	82	0.00053	83	-0.00003	84	-0.00026	85	0.00000	86	-0.00000	87	0.00002	88	0.00001
89	0.00008	90	0.00016	91	-0.00017	92	0.00107	93	-0.00350	94	0.02705	95	0.02574	96	0.01440
97	0.02547	98	-0.00337	99	0.00102	100	0.01233	101	-0.00016	102	0.00007	103	0.00015	104	-0.00023
105	0.00006	106	0.00039	107	-0.00268	108	-0.15530	109	-0.28443	110	-0.16158	111	-0.28210	112	-0.16863
113	0.01270	114	-0.00267	115	-0.13726	116	-0.29390	117	0.01661	118	0.00040	119	-0.00023	120	0.00006
121	0.00007	122	-0.00002	123	0.00031	124	0.00043	125	0.02686	126	-0.00304	127	0.00087	128	1.88724
129	-0.32137	130	0.00044	131	0.00031	132	-0.00002	133	0.00007	134	-0.00005	135	0.00000	136	-0.00003
137	-0.00019	138	-0.00013	139	-0.00019	140	-0.00002	141	-0.00005	142	0.00000	143	0.00001	144	0.00000
145	0.00003	146	0.00003	147	0.00006	148	0.00003	149	0.00003	150	0.00000	151	0.00001	152	-0.00001
153	0.00000	154	-0.00001	155	-0.00003	156	-0.00002	157	-0.00003	158	-0.00001	159	-0.00001	160	-0.00000

605. SECONDS

0.0070 DAYS

TIME

INTERVAL FROM		0.0070 TO		0.0210 DAYS	
WELL PW 1	NODE	1	DISCHARGE	1.3380	
1	0.00000	2	0.00000	3	-0.00000
9	0.00000	10	0.00000	11	-0.00000
17	0.00000	18	0.00000	19	0.00001
25	-0.00001	26	-0.00000	27	-0.00001
33	0.00000	34	0.00002	35	0.00000
41	0.00000	42	-0.00000	43	0.00000
49	-0.00028	50	-0.00016	51	0.00005
57	-0.00001	58	0.00005	59	-0.00000
65	0.00108	66	0.00059	67	-0.00003
73	-0.00001	74	0.00000	75	-0.00008
81	-0.00377	82	0.00062	83	-0.00004
89	0.00010	90	0.00018	91	-0.00020
97	0.02973	98	-0.00385	99	0.00118
105	0.00007	106	0.00045	107	-0.00312
113	0.01510	114	-0.00316	115	-0.16927
121	0.00008	122	-0.00003	123	0.00036
129	-0.37064	130	0.00053	131	0.00036
137	-0.00022	138	-0.00015	139	-0.00022
145	0.00004	146	0.00003	147	0.00007
153	0.00000	154	-0.00001	155	-0.00003
				156	-0.00002
				157	-0.00003
				158	-0.00001
				159	-0.00002
				160	0.00000
				161	0.00000
				162	0.00000
				163	0.00000
				164	0.00000
				165	0.00000
				166	0.00000
				167	0.00000
				168	0.00000
				169	0.00000
				170	0.00000
				171	0.00000
				172	0.00000
				173	0.00000
				174	0.00000
				175	0.00000
				176	0.00000
				177	0.00000
				178	0.00000
				179	0.00000
				180	0.00000
				181	0.00000
				182	0.00000
				183	0.00000
				184	0.00000
				185	0.00000
				186	0.00000
				187	0.00000
				188	0.00000
				189	0.00000
				190	0.00000
				191	0.00000
				192	0.00000
				193	0.00000
				194	0.00000
				195	0.00000
				196	0.00000
				197	0.00000
				198	0.00000
				199	0.00000
				200	0.00000

PIEZOMETRIC SURFACE (NODE AND DRAWDOWN)

TIME		0.0093 DAYS		806. SECONDS	
1	0.00000	2	0.00000	3	-0.00000
9	0.00000	10	0.00000	11	-0.00000
17	0.00000	18	0.00000	19	0.00001
25	-0.00001	26	-0.00000	27	-0.00001
33	0.00000	34	0.00002	35	0.00000
41	0.00000	42	-0.00000	43	0.00000
49	-0.00032	50	-0.00018	51	0.00005
57	-0.00001	58	0.00006	59	-0.00000
65	0.00125	66	0.00070	67	-0.00004
73	-0.00001	74	0.00000	75	-0.00009
81	-0.00448	82	0.00074	83	-0.00005
89	0.00011	90	0.00021	91	-0.00022
97	0.03512	98	-0.00437	99	0.00138
105	0.00008	106	0.00053	107	-0.00365
113	0.01784	114	-0.00379	115	-0.21835
				116	-0.39384
				117	0.02745
				118	0.00058
				119	-0.00031
				120	0.00008
				121	0.00000
				122	0.00000
				123	0.00000
				124	0.00000
				125	0.00000
				126	0.00000
				127	0.00000
				128	0.00000
				129	0.00000
				130	0.00000
				131	0.00000
				132	0.00000
				133	0.00000
				134	0.00000
				135	0.00000
				136	0.00000
				137	0.00000
				138	0.00000
				139	0.00000
				140	0.00000
				141	0.00000
				142	0.00000
				143	0.00000
				144	0.00000
				145	0.00000
				146	0.00000
				147	0.00000
				148	0.00000
				149	0.00000
				150	0.00000
				151	0.00000
				152	0.00000
				153	0.00000
				154	0.00000
				155	0.00000
				156	0.00000
				157	0.00000
				158	0.00000
				159	0.00000
				160	0.00000
				161	0.00000
				162	0.00000
				163	0.00000
				164	0.00000
				165	0.00000
				166	0.00000
				167	0.00000
				168	0.00000
				169	0.00000
				170	0.00000
				171	0.00000
				172	0.00000
				173	0.00000
				174	0.00000
				175	0.00000
				176	0.00000
				177	0.00000
				178	0.00000
				179	0.00000
				180	0.00000
				181	0.00000
				182	0.00000
				183	0.00000
				184	0.00000
				185	0.00000
				186	0.00000
				187	0.00000
				188	0.00000
				189	0.00000
				190	0.00000
				191	0.00000
				192	0.00000
				193	0.00000
				194	0.00000
				195	0.00000
				196	0.00000
				197	0.00000
				198	0.00000
				199	0.00000
				200	0.00000

121	0.00009	122	-0.00003	123	0.00042	124	0.00059	125	0.03588	126	-0.00364	127	0.00113	128	2.95092
129	-0.42940	130	0.00064	131	0.00043	132	-0.00003	133	0.00009	134	-0.00006	135	0.00001	136	-0.00003
137	-0.00025	138	-0.00016	139	-0.00025	140	-0.00004	141	-0.00006	142	0.00000	143	0.00002	144	0.00000
145	0.00004	146	0.00004	147	0.00008	148	0.00004	149	0.00004	150	0.00000	151	0.00002	152	-0.00002
153	0.00000	154	-0.00001	155	-0.00004	156	-0.00002	157	-0.00004	158	-0.00001	159	-0.00002	160	0.00000

TIME		0.0117 DAYS		1008. SECONDS											
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00001	28	-0.00003	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00003	35	0.00000	36	0.00007	37	0.00005	38	-0.00002	39	0.00013	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00003	45	0.00000	46	-0.00010	47	0.00001	48	-0.00006
49	-0.00036	50	-0.00019	51	0.00006	52	-0.00038	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00001	58	0.00007	59	-0.00000	60	-0.00000	61	0.00014	62	-0.00006	63	0.00063	64	0.00086
65	0.00135	66	0.00078	67	-0.00005	68	0.00060	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00000	75	-0.00010	76	-0.00043	77	0.00012	78	0.00093	79	-0.00549	80	-0.00539
81	-0.00499	82	0.00082	83	-0.00005	84	-0.00039	85	0.00000	86	-0.00000	87	0.00002	88	0.00001
89	0.00012	90	0.00023	91	-0.00024	92	0.00157	93	-0.00483	94	0.04056	95	0.03911	96	0.02929
97	0.03901	98	-0.00467	99	0.00151	100	0.02137	101	-0.00022	102	0.00011	103	0.00022	104	-0.00034
105	0.00009	106	0.00057	107	-0.00399	108	-0.29235	109	-0.39132	110	-0.31754	111	-0.39001	112	-0.31387
113	0.01928	114	-0.00425	115	-0.26263	116	-0.43100	117	0.03320	118	0.00065	119	-0.00035	120	0.00009
121	0.00010	122	-0.00003	123	0.00046	124	0.00064	125	0.03900	126	-0.00365	127	0.00121	128	3.55348
129	-0.46901	130	0.00073	131	0.00048	132	-0.00004	133	0.00010	134	-0.00007	135	0.00001	136	-0.00004
137	-0.00027	138	-0.00017	139	-0.00028	140	-0.00004	141	-0.00007	142	0.00001	143	0.00002	144	0.00000
145	0.00005	146	0.00004	147	0.00009	148	0.00004	149	0.00005	150	0.00000	151	0.00002	152	-0.00002
153	0.00000	154	-0.00001	155	-0.00004	156	-0.00002	157	-0.00004	158	-0.00001	159	-0.00002	160	0.00000

TIME		0.0140 DAYS		1210. SECONDS											
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00001	28	-0.00003	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00003	35	0.00001	36	0.00007	37	0.00005	38	-0.00002	39	0.00013	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00003	45	0.00000	46	-0.00011	47	0.00001	48	-0.00006
49	-0.00038	50	-0.00020	51	0.00006	52	-0.00040	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00001	58	0.00007	59	-0.00000	60	-0.00000	61	0.00014	62	-0.00006	63	0.00068	64	0.00093
65	0.00140	66	0.00083	67	-0.00005	68	0.00063	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00000	75	-0.00010	76	-0.00045	77	0.00012	78	0.00100	79	-0.00594	80	-0.00533
81	-0.00531	82	0.00086	83	-0.00006	84	-0.00041	85	0.00000	86	-0.00000	87	0.00003	88	0.00001
89	0.00013	90	0.00024	91	-0.00025	92	0.00164	93	-0.00494	94	0.04294	95	0.04162	96	0.03361
97	0.04162	98	-0.00478	99	0.00158	100	0.02301	101	-0.00023	102	0.00011	103	0.00022	104	-0.00035
105	0.00009	106	0.00059	107	-0.00417	108	-0.33427	109	-0.39959	110	-0.36858	111	-0.39948	112	-0.35723
113	0.01943	114	-0.00456	115	-0.30178	116	-0.45382	117	0.03803	118	0.00070	119	-0.00036	120	0.00010
121	0.00010	122	-0.00004	123	0.00047	124	0.00067	125	0.04065	126	-0.00346	127	0.00123	128	4.11621
129	-0.49286	130	0.00079	131	0.00051	132	-0.00004	133	0.00011	134	-0.00007	135	0.00001	136	-0.00004

137	-0.00027	138	-0.00017	139	-0.00029	140	-0.00004	141	-0.00008	142	0.00001	143	0.00002	144	0.00000
145	0.00005	146	0.00004	147	0.00009	148	0.00004	149	0.00005	150	0.00000	151	0.00002	152	-0.00002
153	0.00001	154	-0.00001	155	-0.00004	156	-0.00002	157	-0.00004	158	-0.00001	159	-0.00002	160	0.00000

TIME 0.0163 DAYS 1411. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00001	28	-0.00003	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00003	35	0.00001	36	0.00007	37	0.00005	38	-0.00002	39	0.00013	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00003	45	0.00000	46	-0.00012	47	0.00001	48	-0.00007
49	-0.00038	50	-0.00019	51	0.00006	52	-0.00040	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00001	58	0.00007	59	-0.00000	60	-0.00000	61	0.00015	62	-0.00007	63	0.00071	64	0.00097
65	0.00140	66	0.00085	67	-0.00006	68	0.00065	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00001	75	-0.00010	76	-0.00046	77	0.00012	78	0.00105	79	-0.00622	80	-0.00508
81	-0.00547	82	0.00088	83	-0.00006	84	-0.00041	85	0.00000	86	-0.00000	87	0.00003	88	0.00001
89	0.00013	90	0.00024	91	-0.00025	92	0.00165	93	-0.00490	94	0.04415	95	0.04300	96	0.03699
97	0.04310	98	-0.00474	99	0.00160	100	0.02363	101	-0.00023	102	0.00011	103	0.00023	104	-0.00035
105	0.00009	106	0.00059	107	-0.00420	108	-0.37012	109	-0.39504	110	-0.41421	111	-0.39643	112	-0.39369
113	0.01836	114	-0.00473	115	-0.33568	116	-0.46479	117	0.04186	118	0.00073	119	-0.00037	120	0.00010
121	0.00010	122	-0.00004	123	0.00048	124	0.00067	125	0.04104	126	-0.00313	127	0.00121	128	4.64273
129	-0.50372	130	0.00082	131	0.00053	132	-0.00004	133	0.00011	134	-0.00007	135	0.00001	136	-0.00004
137	-0.00027	138	-0.00016	139	-0.00029	140	-0.00005	141	-0.00008	142	0.00001	143	0.00002	144	0.00000
145	0.00005	146	0.00004	147	0.00009	148	0.00004	149	0.00005	150	0.00000	151	0.00002	152	-0.00002
153	0.00001	154	-0.00001	155	-0.00004	156	-0.00002	157	-0.00004	158	-0.00001	159	-0.00002	160	0.00000

TIME 0.0187 DAYS 1613. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00001	28	-0.00003	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00003	35	0.00001	36	0.00007	37	0.00005	38	-0.00002	39	0.00013	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00003	45	0.00000	46	-0.00012	47	0.00001	48	-0.00007
49	-0.00038	50	-0.00018	51	0.00006	52	-0.00039	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00001	58	0.00007	59	-0.00000	60	-0.00000	61	0.00015	62	-0.00007	63	0.00073	64	0.00099
65	0.00136	66	0.00085	67	-0.00006	68	0.00065	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00001	75	-0.00010	76	-0.00046	77	0.00012	78	0.00107	79	-0.00634	80	-0.00468
81	-0.00548	82	0.00087	83	-0.00006	84	-0.00041	85	0.00000	86	-0.00000	87	0.00002	88	0.00001
89	0.00013	90	0.00023	91	-0.00024	92	0.00162	93	-0.00472	94	0.04438	95	0.04343	96	0.03941
97	0.04361	98	-0.00457	99	0.00158	100	0.02327	101	-0.00022	102	0.00011	103	0.00022	104	-0.00034
105	0.00009	106	0.00057	107	-0.00410	108	-0.39998	109	-0.37998	110	-0.45431	111	-0.38313	112	-0.42340
113	0.01618	114	-0.00478	115	-0.36430	116	-0.46591	117	0.04466	118	0.00074	119	-0.00037	120	0.00010
121	0.00010	122	-0.00004	123	0.00046	124	0.00064	125	0.04038	126	-0.00269	127	0.00115	128	5.13627
129	-0.50384	130	0.00083	131	0.00053	132	-0.00004	133	0.00011	134	-0.00007	135	0.00001	136	-0.00004
137	-0.00026	138	-0.00015	139	-0.00029	140	-0.00005	141	-0.00008	142	0.00001	143	0.00002	144	0.00000
145	0.00005	146	0.00003	147	0.00008	148	0.00004	149	0.00005	150	0.00000	151	0.00002	152	-0.00002

153	0.00001	154	-0.00001	155	-0.00004	156	-0.00002	157	-0.00004	158	-0.00001	159	-0.00002	160	0.00000
TIME															
0.0210 DAYS															
1814. SECONDS															
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00001	28	-0.00003	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00003	35	0.00001	36	0.00007	37	0.00005	38	-0.00002	39	0.00012	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00002	45	0.00000	46	-0.00011	47	0.00001	48	-0.00007
49	-0.00037	50	-0.00017	51	0.00005	52	-0.00037	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00001	58	0.00006	59	-0.00000	60	-0.00000	61	0.00014	62	-0.00007	63	0.00073	64	0.00098
65	0.00129	66	0.00082	67	-0.00006	68	0.00064	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00001	75	-0.00010	76	-0.00045	77	0.00012	78	0.00106	79	-0.00633	80	-0.00416
81	-0.00537	82	0.00084	83	-0.00005	84	-0.00039	85	0.00000	86	-0.00000	87	0.00002	88	0.00001
89	0.00012	90	0.00022	91	-0.00023	92	0.00155	93	-0.00445	94	0.04376	95	0.04303	96	0.04087
97	0.04328	98	-0.00430	99	0.00152	100	0.02200	101	-0.00021	102	0.00010	103	0.00021	104	-0.00032
105	0.00008	106	0.00053	107	-0.00390	108	-0.42400	109	-0.35631	110	-0.48891	111	-0.36139	112	-0.44660
113	0.01302	114	-0.00471	115	-0.38773	116	-0.45884	117	0.04646	118	0.00073	119	-0.00035	120	0.00009
121	0.00009	122	-0.00004	123	0.00044	124	0.00060	125	0.03882	126	-0.00216	127	0.00106	128	5.59968
129	-0.49507	130	0.00083	131	0.00052	132	-0.00004	133	0.00010	134	-0.00006	135	0.00001	136	-0.00004
137	-0.00024	138	-0.00013	139	-0.00027	140	-0.00005	141	-0.00007	142	0.00001	143	0.00002	144	0.00000
145	0.00004	146	0.00003	147	0.00008	148	0.00004	149	0.00005	150	0.00000	151	0.00002	152	-0.00001
153	0.00001	154	-0.00001	155	-0.00003	156	-0.00002	157	-0.00004	158	-0.00001	159	-0.00002	160	0.00000

INTERVAL FROM 0.0210 TO 0.1670 DAYS

WELL PW 1 NODE 1 DISCHARGE 1.3380

PIEZOMETRIC SURFACE (NODE AND DRAWDOWN)

TIME															
0.0419 DAYS															
3616. SECONDS															
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	-0.00000	6	0.00000	7	0.00000	8	0.00000
9	0.00000	10	-0.00000	11	-0.00000	12	-0.00000	13	0.00000	14	-0.00000	15	-0.00000	16	-0.00000
17	0.00000	18	0.00000	19	0.00000	20	0.00000	21	-0.00000	22	0.00000	23	0.00000	24	-0.00000
25	-0.00000	26	0.00000	27	-0.00000	28	-0.00000	29	0.00000	30	-0.00000	31	0.00000	32	0.00000
33	-0.00000	34	0.00001	35	0.00000	36	0.00001	37	0.00001	38	-0.00000	39	0.00001	40	0.00000
41	-0.00000	42	0.00000	43	0.00000	44	-0.00000	45	-0.00000	46	-0.00004	47	0.00001	48	-0.00003
49	-0.00007	50	-0.00001	51	0.00000	52	-0.00005	53	0.00000	54	0.00000	55	0.00000	56	-0.00000
57	-0.00000	58	0.00001	59	0.00000	60	0.00000	61	0.00003	62	-0.00003	63	0.00029	64	0.00030
65	0.00017	66	0.00016	67	-0.00002	68	0.00017	69	-0.00000	70	-0.00000	71	0.00000	72	0.00000
73	-0.00000	74	0.00000	75	-0.00002	76	-0.00009	77	0.00001	78	0.00034	79	-0.00242	80	0.00078

81	-0.00144	82	0.00016	83	-0.00001	84	-0.00005	85	-0.00000	86	0.00000	87	0.00000	88	-0.00000
89	0.00001	90	0.00002	91	-0.00002	92	0.00026	93	-0.00045	94	0.01887	95	0.01837	96	0.01563
97	0.01617	98	-0.00011	99	0.00020	100	-0.01190	101	-0.00001	102	0.00000	103	0.00001	104	-0.00002
105	0.00000	106	-0.00000	107	-0.00041	108	-0.35004	109	0.07623	110	-0.50111	111	0.05724	112	-0.34770
113	-0.02864	114	-0.00101	115	-0.32670	116	-0.15423	117	0.02120	118	0.00012	119	-0.00004	120	0.00001
121	0.00000	122	-0.00001	123	0.00004	124	0.00002	125	0.00586	126	0.00197	127	-0.00008	128	8.36872
129	-0.16695	130	0.00016	131	0.00010	132	-0.00002	133	0.00001	134	0.00000	135	0.00000	136	-0.00000
137	0.00001	138	0.00004	139	-0.00001	140	-0.00001	141	-0.00001	142	0.00000	143	-0.00000	144	-0.00000
145	-0.00000	146	-0.00001	147	-0.00001	148	-0.00000	149	0.00000	150	-0.00000	151	0.00000	152	0.00000
153	0.00000	154	0.00000	155	0.00000	156	0.00000	157	0.00000	158	-0.00000	159	0.00000	160	0.00000

TIME		0.0627 DAYS		5419. SECONDS		5419. SECONDS		0.0836 DAYS		7221. SECONDS		7221. SECONDS	
1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000
9	-0.00000	10	-0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000
17	-0.00000	18	-0.00000	19	-0.00000	20	0.00000	21	-0.00000	22	0.00000	23	-0.00000
25	0.00001	26	0.00000	27	0.00000	28	0.00002	29	0.00000	30	-0.00000	31	0.00000
33	-0.00000	34	-0.00001	35	-0.00000	36	-0.00003	37	-0.00002	38	0.00001	39	-0.00006
41	-0.00000	42	0.00000	43	-0.00000	44	0.00001	45	-0.00000	46	0.00003	47	-0.00000
49	0.00016	50	0.00009	51	-0.00003	52	0.00019	53	0.00000	54	-0.00000	55	0.00000
57	0.00000	58	-0.00003	59	0.00000	60	0.00000	61	-0.00006	62	0.00001	63	-0.00014
65	-0.00061	66	-0.00037	67	0.00002	68	-0.00024	69	-0.00000	70	-0.00000	71	0.00000
73	0.00001	74	-0.00000	75	0.00005	76	0.00019	77	-0.00006	78	-0.00033	79	0.00138
81	0.00191	82	-0.00036	83	0.00002	84	0.00021	85	-0.00000	86	0.00000	87	-0.00001
89	-0.00007	90	-0.00011	91	0.00012	92	-0.00070	93	0.00245	94	-0.00387	95	-0.00579
97	-0.01362	98	0.00319	99	-0.00087	100	-0.03946	101	0.00014	102	-0.00006	103	-0.00013
105	-0.00004	106	-0.00031	107	0.00191	108	-0.11847	109	0.56271	110	-0.33952	111	0.53634
113	-0.05225	114	0.00253	115	-0.11115	116	0.23486	117	-0.01578	118	-0.00044	119	0.00022
121	-0.00005	122	0.00001	123	-0.00023	124	-0.00032	125	-0.02231	126	0.00350	127	-0.00078
129	0.23955	130	-0.00049	131	-0.00028	132	0.00002	133	-0.00006	134	0.00004	135	-0.00000
137	0.00016	138	0.00012	139	0.00019	140	0.00002	141	0.00004	142	-0.00000	143	-0.00001
145	-0.00003	146	-0.00002	147	-0.00006	148	-0.00003	149	-0.00003	150	-0.00000	151	-0.00001
153	-0.00000	154	0.00001	155	0.00003	156	0.00002	157	0.00003	158	0.00001	159	0.00001

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	-0.00000	18	-0.00000	19	-0.00000	20	0.00000	21	-0.00000	22	0.00000	23	-0.00000	24	-0.00000
25	0.00001	26	0.00000	27	0.00000	28	0.00002	29	0.00000	30	-0.00000	31	0.00000	32	-0.00001
33	-0.00000	34	-0.00002	35	-0.00000	36	-0.00004	37	-0.00003	38	0.00001	39	-0.00008	40	0.00000
41	-0.00000	42	0.00000	43	-0.00000	44	0.00002	45	-0.00000	46	0.00006	47	-0.00001	48	0.00004
49	0.00023	50	0.00011	51	-0.00004	52	0.00025	53	0.00000	54	-0.00000	55	0.00000	56	-0.00000
57	0.00001	58	-0.00005	59	0.00000	60	0.00000	61	-0.00009	62	0.00004	63	-0.00036	64	-0.00060
65	-0.00083	66	-0.00056	67	0.00004	68	-0.00041	69	-0.00000	70	-0.00000	71	0.00000	72	-0.00000
73	0.00001	74	-0.00000	75	0.00007	76	0.00028	77	-0.00008	78	-0.00063	79	0.00327	80	0.00360
81	0.00332	82	-0.00052	83	0.00004	84	0.00030	85	-0.00000	86	0.00000	87	-0.00002	88	-0.00001
89	-0.00009	90	-0.00014	91	0.00015	92	-0.00098	93	0.00344	94	-0.01495	95	-0.01917	96	-0.04677

97	-0.03542	98	0.00467	99	-0.00132	100	-0.04841	101	0.00019	102	-0.00008	103	-0.00018	104	0.00020
105	-0.00005	106	-0.00029	107	0.00224	108	0.18932	109	1.04276	110	-0.08670	111	1.01390	112	0.23930
113	-0.04790	114	0.00442	115	0.18330	116	0.64434	117	-0.04786	118	-0.00072	119	0.00032	120	-0.00009
121	-0.00006	122	0.00002	123	-0.00027	124	-0.00033	125	-0.03556	126	0.00256	127	-0.00087	128	11.76483
129	0.65809	130	-0.00082	131	-0.00048	132	0.00004	133	-0.00010	134	0.00004	135	-0.00001	136	0.00002
137	0.00017	138	0.00011	139	0.00025	140	0.00005	141	0.00007	142	-0.00001	143	-0.00001	144	-0.00000
145	-0.00003	146	-0.00002	147	-0.00006	148	-0.00004	149	-0.00004	150	-0.00000	151	-0.00002	152	0.00001
153	-0.00000	154	0.00001	155	0.00002	156	0.00002	157	0.00003	158	0.00001	159	0.00002	160	-0.00000

TIME 0.1044 DAYS 9023. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	-0.00000	18	-0.00000	19	-0.00000	20	0.00000	21	-0.00000	22	0.00000	23	-0.00000	24	-0.00000
25	0.00001	26	0.00000	27	0.00000	28	0.00001	29	0.00000	30	-0.00000	31	0.00000	32	-0.00000
33	-0.00000	34	-0.00001	35	-0.00000	36	-0.00003	37	-0.00002	38	0.00001	39	-0.00005	40	0.00000
41	-0.00000	42	0.00000	43	-0.00000	44	0.00001	45	-0.00000	46	0.00004	47	-0.00001	48	0.00004
49	0.00015	50	0.00006	51	-0.00002	52	0.00016	53	0.00000	54	-0.00000	55	0.00000	56	-0.00000
57	0.00000	58	-0.00003	59	0.00000	60	0.00000	61	-0.00006	62	0.00004	63	-0.00029	64	-0.00044
65	-0.00051	66	-0.00040	67	0.00004	68	-0.00032	69	-0.00000	70	-0.00000	71	0.00000	72	-0.00000
73	0.00000	74	-0.00000	75	0.00005	76	0.00018	77	-0.00005	78	-0.00048	79	0.00260	80	0.00196
81	0.00257	82	-0.00034	83	0.00002	84	0.00022	85	-0.00000	86	-0.00000	87	-0.00001	88	-0.00000
89	-0.00006	90	-0.00007	91	0.00008	92	-0.00058	93	0.00259	94	-0.01084	95	-0.01755	96	-0.06069
97	-0.04441	98	0.00430	99	-0.00109	100	-0.03590	101	0.00014	102	-0.00005	103	-0.00013	104	0.00008
105	-0.00002	106	-0.00002	107	0.00070	108	0.52670	109	1.49540	110	0.20730	111	1.46678	112	0.59263
113	-0.01612	114	0.00415	115	0.51135	116	1.04675	117	-0.06745	118	-0.00063	119	0.00026	120	-0.00007
121	-0.00002	122	0.00001	123	-0.00010	124	-0.00005	125	-0.03054	126	0.00008	127	-0.00042	128	12.88280
129	1.06276	130	-0.00076	131	-0.00044	132	0.00005	133	-0.00008	134	0.00001	135	-0.00000	136	0.00000
137	0.00006	138	0.00003	139	0.00017	140	0.00005	141	0.00006	142	-0.00001	143	-0.00000	144	0.00000
145	-0.00001	146	-0.00000	147	-0.00003	148	-0.00002	149	-0.00003	150	-0.00000	151	-0.00001	152	0.00000
153	-0.00000	154	0.00000	155	0.00001	156	0.00001	157	0.00002	158	0.00001	159	0.00001	160	-0.00000

TIME 0.1253 DAYS 10825. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	-0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00000	26	-0.00000	27	-0.00000	28	-0.00001	29	-0.00000	30	0.00000	31	-0.00000	32	0.00000
33	0.00000	34	0.00001	35	-0.00000	36	0.00001	37	0.00001	38	-0.00000	39	0.00002	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00001	45	0.00000	46	-0.00002	47	-0.00000	48	0.00000
49	-0.00005	50	-0.00004	51	0.00001	52	-0.00005	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00000	58	0.00001	59	-0.00000	60	-0.00000	61	0.00002	62	0.00000	63	0.00007	64	0.00010
65	0.00021	66	0.00004	67	0.00001	68	0.00000	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00000	74	-0.00000	75	-0.00001	76	-0.00008	77	0.00002	78	0.00008	79	-0.00062	80	-0.00049
81	-0.00010	82	0.00011	83	-0.00001	84	0.00000	85	0.00000	86	-0.00000	87	0.00000	88	0.00000
89	0.00001	90	0.00007	91	-0.00006	92	0.00037	93	0.00036	94	0.00895	95	0.00001	96	-0.05872
97	-0.03900	98	0.00243	99	-0.00028	100	-0.00347	101	0.00002	102	0.00001	103	-0.00001	104	-0.00014
105	0.00004	106	0.00042	107	-0.00226	108	0.86835	109	1.91478	110	0.51393	111	1.88760	112	0.94538

113	0.03862	114	0.00176	115	0.84735	116	1.43046	117	-0.07215	118	-0.00021	119	0.00006	120	-0.00001
121	0.00005	122	-0.00002	123	0.00023	124	0.00041	125	-0.00733	126	-0.00279	127	0.00039	128	13.78140
129	1.44424	130	-0.00031	131	-0.00018	132	0.00003	133	-0.00002	134	-0.00003	135	0.00000	136	-0.00002
137	-0.00012	138	-0.00008	139	-0.00001	140	0.00003	141	0.00001	142	-0.00001	143	0.00001	144	0.00000
145	0.00002	146	0.00002	147	0.00003	148	0.00000	149	0.00000	150	-0.00000	151	-0.00000	152	-0.00001
153	0.00000	154	-0.00000	155	-0.00002	156	-0.00001	157	-0.00001	158	0.00000	159	-0.00000	160	-0.00000

TIME		0.1461 DAYS		12627. SECONDS											
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	-0.00001	26	-0.00000	27	-0.00001	28	-0.00003	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	0.00003	35	0.00001	36	0.00007	37	0.00005	38	-0.00001	39	0.00011	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00002	45	0.00000	46	-0.00011	47	0.00001	48	-0.00006
49	-0.00034	50	-0.00016	51	0.00005	52	-0.00033	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00001	58	0.00006	59	-0.00000	60	-0.00000	61	0.00014	62	-0.00006	63	0.00070	64	0.00092
65	0.00118	66	0.00066	67	-0.00004	68	0.00050	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	0.00000	75	-0.00008	76	-0.00045	77	0.00012	78	0.00095	79	-0.00607	80	-0.00278
81	-0.00423	82	0.00071	83	-0.00005	84	-0.00029	85	0.00000	86	-0.00000	87	0.00002	88	0.00001
89	0.00010	90	0.00024	91	-0.00024	92	0.00167	93	-0.00261	94	0.04343	95	0.03272	96	-0.04148
97	-0.01936	98	-0.00039	99	0.00095	100	0.04568	101	-0.00016	102	0.00008	103	0.00016	104	-0.00041
105	0.00011	106	0.00092	107	-0.00607	108	1.20128	109	2.30103	110	0.81778	111	2.27561	112	1.28571
113	0.11099	114	-0.00250	115	1.17749	116	1.79119	117	-0.06226	118	0.00049	119	-0.00025	120	0.00007
121	0.00012	122	-0.00006	123	0.00066	124	0.00094	125	0.03240	126	-0.00498	127	0.00136	128	14.52660
129	1.80020	130	0.00046	131	0.00028	132	-0.00002	133	0.00007	134	-0.00009	135	0.00001	136	-0.00005
137	-0.00033	138	-0.00019	139	-0.00026	140	-0.00002	141	-0.00005	142	0.00000	143	0.00002	144	0.00000
145	0.00006	146	0.00005	147	0.00009	148	0.00004	149	0.00004	150	0.00000	151	0.00002	152	-0.00002
153	0.00001	154	-0.00001	155	-0.00004	156	-0.00002	157	-0.00004	158	-0.00001	159	-0.00002	160	0.00000

TIME		0.1670 DAYS		14429. SECONDS											
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00001	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00001	24	0.00000
25	-0.00002	26	-0.00000	27	-0.00001	28	-0.00005	29	-0.00000	30	0.00000	31	-0.00000	32	0.00002
33	0.00000	34	0.00005	35	0.00001	36	0.00013	37	0.00008	38	-0.00003	39	0.00021	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00004	45	0.00000	46	-0.00023	47	0.00002	48	-0.00015
49	-0.00067	50	-0.00029	51	0.00009	52	-0.00063	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00001	58	0.00011	59	-0.00000	60	-0.00000	61	0.00028	62	-0.00015	63	0.00156	64	0.00191
65	0.00227	66	0.00136	67	-0.00010	68	0.00111	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00002	74	0.00001	75	-0.00017	76	-0.00087	77	0.00021	78	0.00205	79	-0.01334	80	-0.00408
81	-0.00933	82	0.00138	83	-0.00009	84	-0.00063	85	0.00000	86	-0.00000	87	0.00004	88	0.00001
89	0.00020	90	0.00043	91	-0.00044	92	0.00314	93	-0.00571	94	0.09091	95	0.07899	96	-0.01070
97	0.01338	98	-0.00357	99	0.00242	100	0.10807	101	-0.00034	102	0.00016	103	0.00035	104	-0.00069
105	0.00018	106	0.00141	107	-0.01021	108	1.51944	109	2.65651	110	1.11114	111	2.63276	112	1.60860
113	0.19617	114	-0.00824	115	1.49490	116	2.12819	117	-0.03926	118	0.00138	119	-0.00061	120	0.00017
121	0.00020	122	-0.00011	123	0.00111	124	0.00144	125	0.08637	126	-0.00561	127	0.00235	128	15.16024

129	2.13129	130	0.00145	131	0.00090	132	-0.00008	133	0.00018	134	-0.00015	135	0.00003	136	-0.00008
137	-0.00053	138	-0.00029	139	-0.00054	140	-0.00008	141	-0.00013	142	0.00001	143	0.00003	144	0.00000
145	0.00009	146	0.00007	147	0.00016	148	0.00008	149	0.00009	150	0.00001	151	0.00004	152	-0.00003
153	0.00001	154	-0.00002	155	-0.00007	156	-0.00004	157	-0.00007	158	-0.00002	159	-0.00003	160	0.00001

INTERVAL FROM 0.1670 TO 1.0000 DAYS

WELL PW 1 NODE 1 DISCHARGE 1.3380

PIEZOMETRIC SURFACE (NODE AND DRAWDOWN)

TIME		0.2086 DAYS		18027. SECONDS		0.0000		0.0000		0.0000		0.0000		0.0000	
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00001	19	0.00002	20	-0.00000	21	0.00000	22	-0.00000	23	0.00001	24	0.00000
25	-0.00004	26	-0.00000	27	-0.00002	28	-0.00009	29	-0.00000	30	0.00000	31	-0.00000	32	0.00002
33	0.00000	34	0.00011	35	0.00003	36	0.00025	37	0.00014	38	-0.00004	39	0.00037	40	-0.00000
41	0.00000	42	-0.00000	43	0.00001	44	-0.00006	45	0.00000	46	-0.00053	47	0.00006	48	-0.00037
49	-0.00129	50	-0.00053	51	0.00014	52	-0.00113	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00002	58	0.00018	59	-0.00000	60	-0.00000	61	0.00053	62	-0.00037	63	0.00381	64	0.00386
65	0.00429	66	0.00254	67	-0.00024	68	0.00242	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00003	74	0.00002	75	-0.00032	76	-0.00164	77	0.00034	78	0.00435	79	-0.03177	80	0.00105
81	-0.02026	82	0.00244	83	-0.00015	84	-0.00120	85	0.00000	86	0.00000	87	0.00006	88	0.00001
89	0.00034	90	0.00069	91	-0.00071	92	0.00575	93	-0.00870	94	0.22332	95	0.21071	96	0.08904
97	0.11876	98	-0.00776	99	0.00537	100	0.26415	101	-0.00065	102	0.00026	103	0.00067	104	-0.00107
105	0.00026	106	0.00197	107	-0.01681	108	2.08760	109	3.26719	110	1.64080	111	3.24594	112	2.18081
113	0.39194	114	-0.02272	115	2.06599	116	2.71717	117	0.04416	118	0.00340	119	-0.00134	120	0.00035
121	0.00029	122	-0.00019	123	0.00181	124	0.00193	125	0.23327	126	0.00311	127	0.00369	128	16.14816
129	2.70873	130	0.00366	131	0.00246	132	-0.00027	133	0.00040	134	-0.00021	135	0.00006	136	-0.00010
137	-0.00075	138	-0.00034	139	-0.00103	140	-0.00022	141	-0.00031	142	0.00006	143	0.00004	144	-0.00000
145	0.00011	146	0.00007	147	0.00024	148	0.00013	149	0.00018	150	0.00001	151	0.00007	152	-0.00003
153	0.00002	154	-0.00002	155	-0.00009	156	-0.00005	157	-0.00012	158	-0.00004	159	-0.00006	160	0.00002

TIME		0.2503 DAYS		21626. SECONDS		0.0000		0.0000		0.0000		0.0000		0.0000	
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00001	19	0.00002	20	-0.00000	21	0.00000	22	-0.00000	23	0.00001	24	0.00000
25	-0.00005	26	0.00000	27	-0.00003	28	-0.00011	29	0.00000	30	0.00000	31	-0.00000	32	0.00002
33	-0.00000	34	0.00015	35	0.00005	36	0.00034	37	0.00016	38	-0.00004	39	0.00047	40	-0.00000
41	0.00000	42	-0.00000	43	0.00001	44	-0.00006	45	0.00000	46	-0.00083	47	0.00009	48	-0.00056
49	-0.00175	50	-0.00071	51	0.00016	52	-0.00143	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001

57	-0.00002	58	0.00021	59	0.00000	60	-0.00000	61	0.00070	62	-0.00057	63	0.00629	64	0.00528
65	0.00594	66	0.00317	67	-0.00034	68	0.00352	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00003	74	0.00003	75	-0.00042	76	-0.00216	77	0.00038	78	0.00627	79	-0.05143	80	0.01555
81	-0.02966	82	0.00296	83	-0.00017	84	-0.00155	85	0.00000	86	0.00000	87	0.00006	88	-0.00000
89	0.00040	90	0.00079	91	-0.00081	92	0.00744	93	-0.00629	94	0.38123	95	0.36900	96	0.21412
97	0.25263	98	-0.00736	99	0.00770	100	0.43726	101	-0.00080	102	0.00028	103	0.00086	104	-0.00118
105	0.00028	106	0.00191	107	-0.01984	108	2.59379	109	3.79921	110	2.11650	111	3.77976	112	2.68834
113	0.59811	114	-0.03786	115	2.57743	116	3.23598	117	0.15264	118	0.00523	119	-0.00189	120	0.00047
121	0.00029	122	-0.00023	123	0.00208	124	0.00171	125	0.40525	126	0.02315	127	0.00418	128	16.95023
129	3.21781	130	0.00557	131	0.00408	132	-0.00047	133	0.00056	134	-0.00021	135	0.00008	136	-0.00008
137	-0.00074	138	-0.00028	139	-0.00133	140	-0.00036	141	-0.00047	142	0.00011	143	0.00003	144	-0.00001
145	0.00010	146	0.00004	147	0.00025	148	0.00015	149	0.00023	150	0.00001	151	0.00009	152	-0.00002
153	0.00002	154	-0.00001	155	-0.00007	156	-0.00005	157	-0.00013	158	-0.00005	159	-0.00007	160	0.00003

TIME		0.2919 DAYS		25224. SECONDS											
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	-0.00000	11	-0.00000	12	-0.00000	13	0.00000	14	0.00000	15	-0.00000	16	-0.00000
17	0.00000	18	0.00001	19	0.00002	20	-0.00000	21	0.00000	22	-0.00000	23	0.00001	24	-0.00000
25	-0.00006	26	0.00000	27	-0.00003	28	-0.00011	29	0.00000	30	0.00000	31	-0.00000	32	0.00002
33	-0.00000	34	0.00018	35	0.00006	36	0.00038	37	0.00016	38	-0.00004	39	0.00050	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00005	45	0.00000	46	-0.00110	47	0.00010	48	-0.00069
49	-0.00203	50	-0.00085	51	0.00016	52	-0.00153	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00001	58	0.00020	59	0.00000	60	0.00000	61	0.00076	62	-0.00071	63	0.00874	64	0.00591
65	0.00722	66	0.00312	67	-0.00038	68	0.00424	69	-0.00000	70	-0.00000	71	-0.00000	72	0.00000
73	-0.00002	74	0.00003	75	-0.00045	76	-0.00236	77	0.00033	78	0.00750	79	-0.07035	80	0.03958
81	-0.03605	82	0.00284	83	-0.00012	84	-0.00163	85	0.00000	86	0.00000	87	0.00005	88	-0.00002
89	0.00036	90	0.00071	91	-0.00074	92	0.00797	93	0.00256	94	0.55532	95	0.54393	96	0.35498
97	0.40556	98	-0.00109	99	0.00909	100	0.61815	101	-0.00080	102	0.00024	103	0.00089	104	-0.00099
105	0.00022	106	0.00119	107	-0.01819	108	3.04658	109	4.26885	110	2.54512	111	4.25070	112	3.14108
113	0.80597	114	-0.05195	115	3.03671	116	3.69726	117	0.27683	118	0.00657	119	-0.00220	120	0.00051
121	0.00020	122	-0.00021	123	0.00183	124	0.00075	125	0.59197	126	0.05431	127	0.00384	128	17.62082
129	3.67129	130	0.00685	131	0.00558	132	-0.00065	133	0.00065	134	-0.00015	135	0.00008	136	0.00000
137	-0.00051	138	-0.00013	139	-0.00142	140	-0.00046	141	-0.00059	142	0.00016	143	0.00001	144	-0.00002
145	0.00005	146	-0.00001	147	0.00020	148	0.00014	149	0.00024	150	0.00001	151	0.00010	152	0.00000
153	0.00001	154	0.00000	155	-0.00004	156	-0.00003	157	-0.00012	158	-0.00005	159	-0.00007	160	0.00004

TIME		0.3336 DAYS		28823. SECONDS											
1	-0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	-0.00000	6	0.00000	7	0.00000	8	0.00000
9	0.00000	10	-0.00000	11	-0.00000	12	-0.00000	13	0.00000	14	-0.00000	15	-0.00000	16	-0.00000
17	0.00000	18	0.00001	19	0.00002	20	0.00000	21	-0.00000	22	-0.00000	23	0.00001	24	-0.00000
25	-0.00005	26	0.00001	27	-0.00002	28	-0.00009	29	0.00000	30	0.00000	31	-0.00000	32	0.00001
33	-0.00000	34	0.00020	35	0.00006	36	0.00038	37	0.00013	38	-0.00002	39	0.00048	40	0.00000
41	-0.00000	42	-0.00000	43	0.00000	44	-0.00003	45	-0.00000	46	-0.00133	47	0.00008	48	-0.00073
49	-0.00213	50	-0.00097	51	0.00013	52	-0.00145	53	0.00000	54	-0.00000	55	0.00000	56	0.00000
57	-0.00000	58	0.00015	59	0.00000	60	0.00000	61	0.00070	62	-0.00075	63	0.01097	64	0.00565
65	0.00828	66	0.00238	67	-0.00033	68	0.00446	69	-0.00000	70	-0.00000	71	0.00000	72	0.00000

73	-0.00001	74	0.00003	75	-0.00041	76	-0.00223	77	0.00020	78	0.00790	79	-0.08719	80	0.07244
81	-0.03854	82	0.00208	83	-0.00001	84	-0.00145	85	-0.00000	86	0.00000	87	0.00003	88	-0.00005
89	0.00025	90	0.00047	91	-0.00051	92	0.00731	93	0.01806	94	0.73873	95	0.72819	96	0.50483
97	0.57033	98	0.01155	99	0.00946	100	0.80114	101	-0.00065	102	0.00014	103	0.00078	104	-0.00053
105	0.00009	106	-0.00016	107	-0.01133	108	3.45446	109	4.68858	110	2.93393	111	4.67132	112	3.54811
113	1.01079	114	-0.06378	115	3.45165	116	4.11147	117	0.41011	118	0.00728	119	-0.00225	120	0.00047
121	0.00003	122	-0.00011	123	0.00102	124	-0.00086	125	0.78617	126	0.09555	127	0.00287	128	18.19449
129	4.07933	130	0.00733	131	0.00683	132	-0.00079	133	0.00066	134	-0.00002	135	0.00005	136	0.00012
137	-0.00012	138	0.00005	139	-0.00131	140	-0.00049	141	-0.00067	142	0.00020	143	-0.00002	144	-0.00002
145	-0.00003	146	-0.00007	147	0.00012	148	0.00010	149	0.00022	150	-0.00000	151	0.00010	152	0.00003
153	-0.00001	154	0.00002	155	0.00001	156	-0.00001	157	-0.00008	158	-0.00004	159	-0.00005	160	0.00005

TIME		0.3752 DAYS		32422. SECONDS											
1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	-0.00000	12	-0.00000	13	0.00000	14	-0.00000	15	-0.00000	16	-0.00000
17	0.00000	18	0.00000	19	0.00001	20	0.00000	21	-0.00000	22	0.00000	23	0.00000	24	-0.00000
25	-0.00004	26	0.00001	27	-0.00001	28	-0.00007	29	0.00000	30	-0.00000	31	0.00000	32	0.00000
33	-0.00000	34	0.00021	35	0.00005	36	0.00034	37	0.00009	38	-0.00001	39	0.00043	40	0.00000
41	-0.00000	42	0.00000	43	-0.00000	44	-0.00000	45	-0.00000	46	-0.00149	47	0.00004	48	-0.00066
49	-0.00211	50	-0.00108	51	0.00009	52	-0.00127	53	0.00000	54	-0.00000	55	0.00000	56	-0.00001
57	0.00001	58	0.00008	59	0.00000	60	0.00000	61	0.00055	62	-0.00068	63	0.01283	64	0.00455
65	0.00931	66	0.00104	67	-0.00021	68	0.00414	69	-0.00000	70	-0.00000	71	0.00000	72	-0.00000
73	0.00001	74	0.00001	75	-0.00030	76	-0.00182	77	0.00001	78	0.00744	79	-0.01017	80	0.11304
81	-0.03665	82	0.00075	83	0.00013	84	-0.00105	85	-0.00000	86	0.00000	87	-0.00001	88	-0.00008
89	0.00009	90	0.00013	91	-0.00017	92	0.00557	93	0.03993	94	0.92656	95	0.91659	96	0.65904
97	0.74158	98	0.03046	99	0.00886	100	0.98279	101	-0.00040	102	0.00000	103	0.00055	104	0.00013
105	-0.00008	106	-0.00205	107	0.00092	108	3.82474	109	5.06786	110	3.28930	111	5.05112	112	3.91705
113	1.21009	114	-0.07260	115	3.82917	116	4.48690	117	0.54796	118	0.00730	119	-0.00205	120	0.00037
121	-0.00018	122	0.00006	123	-0.00034	124	-0.00297	125	0.98289	126	0.14546	127	0.00152	128	18.69444
129	4.44981	130	0.00696	131	0.00774	132	-0.00087	133	0.00059	134	0.00015	135	0.00000	136	0.00026
137	0.00038	138	0.00022	139	-0.00102	140	-0.00046	141	-0.00070	142	0.00023	143	-0.00006	144	-0.00003
145	-0.00012	146	-0.00012	147	0.00000	148	0.00004	149	0.00016	150	-0.00001	151	0.00009	152	0.00005
153	-0.00002	154	0.00003	155	0.00006	156	0.00002	157	-0.00003	158	-0.00002	159	-0.00003	160	0.00005

TIME		0.4169 DAYS		36020. SECONDS											
1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	-0.00000	18	-0.00000	19	0.00000	20	0.00000	21	-0.00000	22	0.00000	23	-0.00000	24	-0.00000
25	-0.00002	26	0.00002	27	0.00000	28	-0.00005	29	0.00000	30	-0.00000	31	0.00000	32	-0.00000
33	-0.00000	34	0.00020	35	0.00003	36	0.00028	37	0.00004	38	0.00001	39	0.00037	40	0.00000
41	-0.00000	42	0.00000	43	-0.00000	44	0.00002	45	-0.00000	46	-0.00159	47	-0.00002	48	-0.00047
49	-0.00202	50	-0.00119	51	0.00005	52	-0.00102	53	0.00000	54	-0.00000	55	0.00000	56	-0.00001
57	0.00002	58	-0.00000	59	0.00000	60	0.00000	61	0.00032	62	-0.00051	63	0.01423	64	0.00270
65	0.01052	66	-0.00077	67	-0.00003	68	0.00324	69	-0.00000	70	-0.00000	71	0.00000	72	-0.00000
73	0.00002	74	-0.00001	75	-0.00013	76	-0.00117	77	-0.00020	78	0.00619	79	-0.11149	80	0.16016
81	-0.03021	82	-0.00101	83	0.00031	84	-0.00048	85	-0.00001	86	0.00000	87	-0.00004	88	-0.00011

89	-0.00011	90	-0.00029	91	0.00024	92	0.00292	93	0.06762	94	1.11539	95	1.10556	96	0.81445
97	0.91551	98	0.05527	99	0.00741	100	1.16111	101	-0.00007	102	-0.00015	103	0.00024	104	0.00094
105	-0.00028	106	-0.00437	107	0.01849	108	4.16346	109	5.41388	110	3.61639	111	5.39732	112	4.25404
113	1.40274	114	-0.07795	115	4.17501	116	4.83004	117	0.68732	118	0.00662	119	-0.00163	120	0.00021
121	-0.00042	122	0.00030	123	-0.00221	124	-0.00541	125	1.17884	126	0.20253	127	0.00003	128	19.13687
129	4.78889	130	0.00577	131	0.00825	132	-0.00087	133	0.00045	134	0.00035	135	-0.00007	136	0.00041
137	0.00091	138	0.00034	139	-0.00063	140	-0.00036	141	-0.00069	142	0.00024	143	-0.00009	144	-0.00002
145	-0.00020	146	-0.00016	147	-0.00012	148	-0.00003	149	0.00009	150	-0.00003	151	0.00007	152	0.00007
153	-0.00004	154	0.00004	155	0.00011	156	0.00005	157	0.00003	158	0.00000	159	-0.00000	160	0.00005

TIME		0.4585 DAYS				39619. SECONDS									
1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	-0.00000	18	-0.00001	19	-0.00001	20	0.00000	21	-0.00000	22	0.00000	23	-0.00000	24	-0.00000
25	-0.00001	26	0.00002	27	0.00001	28	-0.00002	29	0.00000	30	-0.00000	31	0.00000	32	-0.00001
33	-0.00000	34	0.00017	35	-0.00001	36	0.00021	37	-0.00000	38	0.00002	39	0.00030	40	0.00000
41	-0.00000	42	0.00000	43	-0.00001	44	0.00004	45	-0.00000	46	-0.00160	47	-0.00012	48	-0.00020
49	-0.00191	50	-0.00134	51	0.00001	52	-0.00079	53	0.00000	54	-0.00000	55	0.00000	56	-0.00001
57	0.00002	58	-0.00008	59	0.00000	60	0.00000	61	0.00004	62	-0.00023	63	0.01511	64	0.00027
65	0.01210	66	-0.00289	67	0.00020	68	0.00181	69	-0.00000	70	-0.00000	71	0.00000	72	-0.00000
73	0.00004	74	-0.00004	75	0.00008	76	-0.00034	77	-0.00041	78	0.00425	79	-0.11820	80	0.21258
81	-0.01924	82	-0.00307	83	0.00050	84	0.00019	85	-0.00001	86	0.00000	87	-0.00008	88	-0.00013
89	-0.00032	90	-0.00074	91	0.00069	92	-0.00042	93	0.10043	94	1.30287	95	1.29270	96	0.96899
97	1.08942	98	0.08539	99	0.00531	100	1.33498	101	0.00029	102	-0.00030	103	-0.00011	104	0.00183
105	-0.00050	106	-0.00697	107	0.04113	108	4.47546	109	5.73214	110	3.91938	111	5.71542	112	4.56401
113	1.58834	114	-0.07964	115	4.49383	116	5.14596	117	0.82620	118	0.00532	119	-0.00106	120	0.00002
121	-0.00067	122	0.00058	123	-0.00454	124	-0.00797	125	1.37187	126	0.26533	127	-0.00137	128	19.53349
129	5.10141	130	0.00387	131	0.00834	132	-0.00078	133	0.00027	134	0.00058	135	-0.00016	136	0.00055
137	0.00143	138	0.00041	139	-0.00017	140	-0.00018	141	-0.00062	142	0.00023	143	-0.00012	144	-0.00001
145	-0.00027	146	-0.00019	147	-0.00023	148	-0.00010	149	0.00000	150	-0.00005	151	0.00004	152	0.00008
153	-0.00006	154	0.00004	155	0.00015	156	0.00007	157	0.00009	158	0.00003	159	0.00003	160	0.00004

TIME		0.5002 DAYS		43217. SECONDS											
1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	-0.00000	18	-0.00001	19	-0.00001	20	0.00000	21	-0.00000	22	0.00000	23	-0.00001	24	-0.00000
25	0.00001	26	0.00003	27	0.00002	28	-0.00000	29	-0.00000	30	-0.00000	31	0.00000	32	-0.00002
33	0.00000	34	0.00013	35	-0.00005	36	0.00014	37	-0.00003	38	0.00003	39	0.00026	40	0.00000
41	-0.00000	42	0.00000	43	-0.00001	44	0.00006	45	-0.00000	46	-0.00155	47	-0.00022	48	0.00016
49	-0.00184	50	-0.00151	51	-0.00001	52	-0.00060	53	0.00000	54	-0.00000	55	0.00000	56	-0.00002
57	0.00003	58	-0.00015	59	-0.00000	60	0.00000	61	-0.00027	62	0.00013	63	0.01544	64	-0.00257
65	0.01420	66	-0.00516	67	0.00046	68	-0.00014	69	-0.00000	70	-0.00000	71	0.00000	72	-0.00000
73	0.00005	74	-0.00006	75	0.00033	76	0.00058	77	-0.00060	78	0.00174	79	-0.12115	80	0.26920
81	-0.00391	82	-0.00529	83	0.00067	84	0.00091	85	-0.00001	86	-0.00000	87	-0.00012	88	-0.00013
89	-0.00051	90	-0.00118	91	0.00113	92	-0.00425	93	0.13767	94	1.48741	95	1.47643	96	1.12130
97	1.26149	98	0.12018	99	0.00272	100	1.50384	101	0.00064	102	-0.00044	103	-0.00048	104	0.00275

105	-0.00071	106	-0.00974	107	0.06852	108	4.76465	109	6.02688	110	4.20158	111	6.00971	112	4.85095
113	1.76689	114	-0.07761	115	4.78941	116	5.43869	117	0.96332	118	0.00348	119	-0.00037	120	-0.00019
121	-0.00091	122	0.00090	123	-0.00724	124	-0.01048	125	1.56063	126	0.33258	127	-0.00249	128	19.89288
129	5.39118	130	0.00137	131	0.00799	132	-0.00062	133	0.00004	134	0.00082	135	-0.00025	136	0.00065
137	0.00188	138	0.00039	139	0.00030	140	0.00006	141	-0.00052	142	0.00019	143	-0.00014	144	-0.00000
145	-0.00032	146	-0.00019	147	-0.00033	148	-0.00016	149	-0.00009	150	-0.00007	151	0.00001	152	0.00008
153	-0.00007	154	0.00004	155	0.00017	156	0.00009	157	0.00014	158	0.00005	159	0.00005	160	0.00003

TIME 0.5418 DAYS 46816. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	-0.00000	18	-0.00001	19	-0.00001	20	0.00000	21	-0.00000	22	0.00000	23	-0.00001	24	-0.00000
25	0.00002	26	0.00003	27	0.00003	28	0.00001	29	-0.00000	30	-0.00000	31	0.00000	32	-0.00002
33	0.00000	34	0.00007	35	-0.00010	36	0.00008	37	-0.00005	38	0.00003	39	0.00023	40	0.00000
41	-0.00000	42	0.00000	43	-0.00001	44	0.00007	45	-0.00000	46	-0.00141	47	-0.00034	48	0.00057
49	-0.00185	50	-0.00172	51	-0.00002	52	-0.00051	53	0.00000	54	-0.00000	55	0.00000	56	-0.00002
57	0.00003	58	-0.00020	59	-0.00000	60	0.00000	61	-0.00059	62	0.00056	63	0.01521	64	-0.00564
65	0.01697	66	-0.00742	67	0.00071	68	-0.00255	69	-0.00000	70	0.00000	71	0.00000	72	-0.00000
73	0.00005	74	-0.00008	75	0.00061	76	0.00156	77	-0.00077	78	-0.00117	79	-0.12039	80	0.32906
81	0.01554	82	-0.00753	83	0.00082	84	0.00164	85	-0.00001	86	-0.00000	87	-0.00015	88	-0.00012
89	-0.00068	90	-0.00159	91	0.00154	92	-0.00837	93	0.17865	94	1.66798	95	1.65580	96	1.27052
97	1.43049	98	0.15898	99	-0.00015	100	1.66745	101	0.00097	102	-0.00054	103	-0.00084	104	0.00364
105	-0.00090	106	-0.01254	107	0.10028	108	5.03415	109	6.30145	110	4.46566	111	6.28359	112	5.11804
113	1.93861	114	-0.07196	115	5.06484	116	5.71140	117	1.09789	118	0.00119	119	0.00039	120	-0.00039
121	-0.00112	122	0.00124	123	-0.01025	124	-0.01277	125	1.74431	126	0.40316	127	-0.00315	128	20.22152
129	5.66129	130	-0.00157	131	0.00722	132	-0.00039	133	-0.00020	134	0.00105	135	-0.00035	136	0.00071
137	0.00224	138	0.00031	139	0.00073	140	0.00035	141	-0.00037	142	0.00013	143	-0.00015	144	0.00002
145	-0.00035	146	-0.00017	147	-0.00040	148	-0.00020	149	-0.00017	150	-0.00008	151	-0.00002	152	0.00007
153	-0.00009	154	0.00003	155	0.00018	156	0.00010	157	0.00018	158	0.00007	159	0.00008	160	0.00002

TIME 0.5835 DAYS 50414. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	0.00000	11	0.00000	12	0.00000	13	0.00000	14	-0.00000	15	0.00000	16	0.00000
17	-0.00001	18	-0.00001	19	-0.00001	20	0.00000	21	-0.00000	22	0.00000	23	-0.00001	24	-0.00000
25	0.00004	26	0.00004	27	0.00004	28	0.00001	29	-0.00000	30	-0.00000	31	0.00000	32	-0.00002
33	0.00000	34	0.00000	35	-0.00015	36	0.00004	37	-0.00006	38	0.00004	39	0.00025	40	0.00000
41	-0.00000	42	0.00000	43	-0.00001	44	0.00007	45	-0.00000	46	-0.00121	47	-0.00047	48	0.00101
49	-0.00197	50	-0.00197	51	-0.00002	52	-0.00055	53	0.00000	54	-0.00000	55	0.00000	56	-0.00001
57	0.00003	58	-0.00023	59	-0.00000	60	0.00000	61	-0.00089	62	0.00102	63	0.01443	64	-0.00878
65	0.02051	66	-0.00952	67	0.00094	68	-0.00536	69	-0.00000	70	0.00000	71	0.00000	72	-0.00000
73	0.00005	74	-0.00010	75	0.00090	76	0.00252	77	-0.00088	78	-0.00437	79	-0.11609	80	0.39136
81	0.03883	82	-0.00967	83	0.00092	84	0.00232	85	-0.00001	86	-0.00001	87	-0.00018	88	-0.00009
89	-0.00081	90	-0.00194	91	0.00190	92	-0.01260	93	0.22275	94	1.84396	95	1.83023	96	1.41615
97	1.59567	98	0.20118	99	-0.00315	100	1.82580	101	0.00124	102	-0.00061	103	-0.00115	104	0.00447
105	-0.00107	106	-0.01526	107	0.13600	108	5.28647	109	6.55848	110	4.71376	111	6.53973	112	5.36787
113	2.10383	114	-0.06281	115	5.32256	116	5.96664	117	1.22944	118	-0.00144	119	0.00116	120	-0.00059

121	-0.00129	122	0.00158	123	-0.01349	124	-0.01467	125	1.92248	126	0.47614	127	-0.00324	128	20.52437
129	5.91421	130	-0.00481	131	0.00603	132	-0.00010	133	-0.00046	134	0.00127	135	-0.00045	136	0.00070
137	0.00248	138	0.00015	139	0.00110	140	0.00067	141	-0.00019	142	0.00005	143	-0.00016	144	0.00005
145	-0.00036	146	-0.00013	147	-0.00044	148	-0.00023	149	-0.00025	150	-0.00010	151	-0.00006	152	0.00006
153	-0.00009	154	0.00002	155	0.00017	156	0.00010	157	0.00020	158	0.00008	159	0.00009	160	0.00000

TIME 0.6251 DAYS 54013. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	0.00000	5	-0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	0.00000	11	0.00000	12	0.00000	13	-0.00000	14	-0.00000	15	0.00000	16	0.00000
17	-0.00001	18	-0.00001	19	-0.00001	20	0.00000	21	-0.00000	22	0.00000	23	-0.00001	24	0.00000
25	0.00004	26	0.00004	27	0.00004	28	0.00000	29	-0.00000	30	-0.00000	31	0.00000	32	-0.00002
33	0.00000	34	-0.00008	35	-0.00019	36	0.00002	37	-0.00006	38	0.00003	39	0.00030	40	0.00000
41	-0.00000	42	0.00000	43	-0.00001	44	0.00007	45	0.00000	46	-0.00094	47	-0.00058	48	0.00146
49	-0.00223	50	-0.00225	51	0.00000	52	-0.00073	53	0.00000	54	-0.00000	55	0.00000	56	-0.00001
57	0.00003	58	-0.00023	59	-0.00000	60	-0.00000	61	-0.00116	62	0.00151	63	0.01314	64	-0.01185
65	0.02490	66	-0.01132	67	0.00112	68	-0.00850	69	0.00000	70	0.00000	71	0.00000	72	-0.00000
73	0.00005	74	-0.00011	75	0.00119	76	0.00344	77	-0.00095	78	-0.00771	79	-0.10843	80	0.45540
81	0.06563	82	-0.01158	83	0.00096	84	0.00293	85	-0.00000	86	-0.00002	87	-0.00020	88	-0.00004
89	-0.00090	90	-0.00222	91	0.00218	92	-0.01678	93	0.26941	94	2.01495	95	1.99941	96	1.55791
97	1.75654	98	0.24622	99	-0.00612	100	1.97897	101	0.00145	102	-0.00064	103	-0.00142	104	0.00519
105	-0.00119	106	-0.01779	107	0.17528	108	5.52373	109	6.80012	110	4.94768	111	6.78030	112	5.60254
113	2.26293	114	-0.05038	115	5.56471	116	6.20653	117	1.35772	118	-0.00429	119	0.00192	120	-0.00076
121	-0.00141	122	0.00190	123	-0.01689	124	-0.01606	125	2.09494	126	0.55078	127	-0.00265	128	20.80530
129	6.15198	130	-0.00821	131	0.00445	132	0.00024	133	-0.00070	134	0.00146	135	-0.00054	136	0.00062
137	0.00259	138	-0.00007	139	0.00136	140	0.00100	141	0.00002	142	-0.00004	143	-0.00015	144	0.00008
145	-0.00034	146	-0.00008	147	-0.00045	148	-0.00024	149	-0.00030	150	-0.00011	151	-0.00009	152	0.00004
153	-0.00010	154	-0.00000	155	0.00015	156	0.00009	157	0.00021	158	0.00009	159	0.00010	160	-0.00001

TIME 0.6668 DAYS 57612. SECONDS

1	0.00000	2	-0.00000	3	0.00000	4	0.00000	5	0.00000	6	-0.00000	7	-0.00000	8	-0.00000
9	-0.00000	10	0.00000	11	0.00000	12	0.00000	13	-0.00000	14	0.00000	15	0.00000	16	0.00000
17	-0.00001	18	-0.00001	19	-0.00001	20	-0.00000	21	-0.00000	22	0.00000	23	-0.00000	24	0.00000
25	0.00005	26	0.00003	27	0.00003	28	-0.00001	29	-0.00000	30	-0.00000	31	0.00000	32	-0.00001
33	0.00000	34	-0.00016	35	-0.00024	36	0.00003	37	-0.00004	38	0.00003	39	0.00039	40	-0.00000
41	-0.00000	42	0.00000	43	-0.00000	44	0.00005	45	0.00000	46	-0.00061	47	-0.00069	48	0.00189
49	-0.00265	50	-0.00258	51	0.00004	52	-0.00106	53	-0.00000	54	-0.00000	55	0.00000	56	-0.00001
57	0.00002	58	-0.00021	59	-0.00000	60	-0.00000	61	-0.00138	62	0.00198	63	0.01136	64	-0.01470
65	0.03019	66	-0.01272	67	0.00124	68	-0.01192	69	0.00000	70	0.00000	71	-0.00000	72	-0.00000
73	0.00004	74	-0.00011	75	0.00147	76	0.00427	77	-0.00097	78	-0.01107	79	-0.09764	80	0.52065
81	0.09562	82	-0.01318	83	0.00093	84	0.00344	85	0.00000	86	-0.00002	87	-0.00021	88	0.00003
89	-0.00093	90	-0.00241	91	0.00238	92	-0.02078	93	0.31816	94	2.18077	95	2.16329	96	1.69568
97	1.91288	98	0.29358	99	-0.00892	100	2.12711	101	0.00159	102	-0.00063	103	-0.00162	104	0.00579
105	-0.00127	106	-0.02003	107	0.21772	108	5.74763	109	7.02809	110	5.16890	111	7.00712	112	5.82386
113	2.41627	114	-0.03487	115	5.79300	116	6.43278	117	1.48263	118	-0.00727	119	0.00263	120	-0.00090
121	-0.00148	122	0.00218	123	-0.02038	124	-0.01684	125	2.26168	126	0.62644	127	-0.00132	128	21.06733
129	6.37629	130	-0.01163	131	0.00252	132	0.00062	133	-0.00093	134	0.00163	135	-0.00060	136	0.00046

137	0.00256	138	-0.00033	139	0.00150	140	0.00134	141	0.00025	142	-0.00014	143	-0.00014	144	0.00012
145	-0.00030	146	-0.00001	147	-0.00043	148	-0.00022	149	-0.00034	150	-0.00013	151	-0.00013	152	0.00002
153	-0.00010	154	-0.00002	155	0.00012	156	0.00007	157	0.00020	158	0.00009	159	0.00011	160	-0.00003
TIME 0.7084 DAYS 61210. SECONDS															
1	0.00000	2	-0.00000	3	0.00000	4	0.00000	5	0.00000	6	-0.00000	7	-0.00000	8	-0.00000
9	-0.00000	10	0.00000	11	0.00000	12	0.00000	13	-0.00000	14	0.00000	15	0.00000	16	0.00000
17	-0.00001	18	-0.00000	19	-0.00000	20	-0.00000	21	0.00000	22	0.00000	23	-0.00000	24	0.00000
25	0.00005	26	0.00003	27	0.00003	28	-0.00004	29	-0.00000	30	0.00000	31	0.00000	32	-0.00000
33	0.00000	34	-0.00024	35	-0.00027	36	0.00007	37	-0.00001	38	0.00002	39	0.00051	40	-0.00000
41	0.00000	42	0.00000	43	-0.00000	44	0.00004	45	0.00000	46	-0.00024	47	-0.00077	48	0.00227
49	-0.00323	50	-0.00294	51	0.00009	52	-0.00155	53	-0.00000	54	0.00000	55	-0.00000	56	-0.00000
57	0.00001	58	-0.00016	59	-0.00000	60	-0.00000	61	-0.00155	62	0.00244	63	0.00914	64	-0.01723
65	0.03643	66	-0.01363	67	0.00128	68	-0.01554	69	0.00000	70	0.00000	71	-0.00000	72	-0.00000
73	0.00003	74	-0.00010	75	0.00174	76	0.00498	77	-0.00094	78	-0.01436	79	-0.08397	80	0.58666
81	0.12847	82	-0.01437	83	0.00082	84	0.00384	85	0.00000	86	-0.00003	87	-0.00020	88	0.00013
89	-0.00092	90	-0.00252	91	0.00249	92	-0.02448	93	0.36857	94	2.34140	95	2.32188	96	1.82943
97	2.06460	98	0.34282	99	-0.01145	100	2.27046	101	0.00164	102	-0.00058	103	-0.00176	104	0.00625
105	-0.00131	106	-0.02190	107	0.26294	108	5.95956	109	7.24384	110	5.37866	111	7.22165	112	6.03326
113	2.56424	114	-0.01654	115	6.00888	116	6.64686	117	1.60417	118	-0.01027	119	0.00326	120	-0.00100
121	-0.00149	122	0.00243	123	-0.02388	124	-0.01690	125	2.42277	126	0.70265	127	0.00081	128	21.31290
129	6.58855	130	-0.01495	131	0.00026	132	0.00101	133	-0.00113	134	0.00175	135	-0.00065	136	0.00021
137	0.00240	138	-0.00063	139	0.00151	140	0.00166	141	0.00050	142	-0.00026	143	-0.00012	144	0.00016
145	-0.00025	146	0.00007	147	-0.00038	148	-0.00019	149	-0.00035	150	-0.00013	151	-0.00017	152	-0.00001
153	-0.00009	154	-0.00005	155	0.00008	156	0.00005	157	0.00018	158	0.00009	159	0.00010	160	-0.00006

1	0.00000	2	0.00000	3	0.00000	4	-0.00000	5	0.00000	6	-0.00000	7	-0.00000	8	-0.00000
9	0.00000	10	0.00000	11	0.00000	12	0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	-0.00000	18	-0.00000	19	0.00000	20	-0.00000	21	0.00000	22	-0.00000	23	0.00000	24	0.00000
25	0.00004	26	0.00002	27	0.00002	28	-0.00007	29	-0.00000	30	0.00000	31	-0.00000	32	0.00000
33	0.00000	34	-0.00033	35	-0.00029	36	0.00013	37	0.00003	38	0.00001	39	0.00067	40	-0.00000
41	0.00000	42	-0.00000	43	-0.00000	44	0.00002	45	0.00000	46	0.00017	47	-0.00083	48	0.00260
49	-0.00398	50	-0.00333	51	0.00015	52	-0.00221	53	-0.00000	54	0.00000	55	-0.00000	56	0.00000
57	-0.00000	58	-0.00009	59	-0.00000	60	-0.00000	61	-0.00165	62	0.00285	63	0.00653	64	-0.01935
65	0.04363	66	-0.01395	67	0.00122	68	-0.01931	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	0.00001	74	-0.00007	75	0.00197	76	0.00557	77	-0.00086	78	-0.01748	79	-0.06766	80	0.65308
81	0.16389	82	-0.01509	83	0.00062	84	0.00412	85	0.00001	86	-0.00004	87	-0.00019	88	0.00024
89	-0.00087	90	-0.00254	91	0.00250	92	-0.02778	93	0.42029	94	2.49687	95	2.47529	96	1.95923
97	2.21170	98	0.39358	99	-0.01361	100	2.40920	101	0.00162	102	-0.00049	103	-0.00183	104	0.00655
105	-0.00129	106	-0.02335	107	0.31059	108	6.16074	109	7.44859	110	5.57806	111	7.42516	112	6.23199
113	2.70717	114	0.00439	115	6.21360	116	6.84995	117	1.72237	118	-0.01322	119	0.00379	120	-0.00107
121	-0.00144	122	0.00261	123	-0.02734	124	-0.01620	125	2.57836	126	0.77900	127	0.00377	128	21.54398
129	6.79001	130	-0.01807	131	-0.00227	132	0.00140	133	-0.00130	134	0.00184	135	-0.00068	136	-0.00013
137	0.00211	138	-0.00095	139	0.00137	140	0.00195	141	0.00077	142	-0.00037	143	-0.00009	144	0.00020
145	-0.00018	146	0.00015	147	-0.00030	148	-0.00014	149	-0.00034	150	-0.00013	151	-0.00020	152	-0.00003

153	-0.00008	154	-0.00007	155	0.00003	156	0.00002	157	0.00014	158	0.00008	159	0.00009	160	-0.00008
TIME															
0.7917 DAYS															
68407. SECONDS															
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	-0.00000	18	0.00000	19	0.00001	20	-0.00000	21	0.00000	22	-0.00000	23	0.00001	24	0.00000
25	0.00003	26	0.00000	27	0.00001	28	-0.00010	29	-0.00000	30	0.00000	31	-0.00000	32	0.00001
33	0.00000	34	-0.00041	35	-0.00030	36	0.00023	37	0.00007	38	0.00000	39	0.00087	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	0.00000	45	0.00000	46	0.00061	47	-0.00085	48	0.00284
49	-0.00491	50	-0.00374	51	0.00021	52	-0.00301	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00001	58	-0.00000	59	-0.00000	60	-0.00000	61	-0.00169	62	0.00320	63	0.00357	64	-0.02098
65	0.05180	66	-0.01365	67	0.00107	68	-0.02316	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00001	74	-0.00004	75	0.00217	76	0.00600	77	-0.00073	78	-0.02035	79	-0.04894	80	0.71962
81	0.20157	82	-0.01529	83	0.00033	84	0.00427	85	0.00001	86	-0.00004	87	-0.00016	88	0.00036
89	-0.00078	90	-0.00247	91	0.00242	92	-0.03060	93	0.47304	94	2.64729	95	2.62369	96	2.08519
97	2.35427	98	0.44551	99	-0.01531	100	2.54356	101	0.00152	102	-0.00037	103	-0.00183	104	0.00671
105	-0.00122	106	-0.02431	107	0.36034	108	6.35219	109	7.64338	110	5.76802	111	7.61871	112	6.42109
113	2.84537	114	0.02768	115	6.40819	116	7.04312	117	1.83733	118	-0.01604	119	0.00422	120	-0.00109
121	-0.00134	122	0.00273	123	-0.03070	124	-0.01469	125	2.72867	126	0.85520	127	0.00757	128	21.76221
129	6.98166	130	-0.02090	131	-0.00505	132	0.00179	133	-0.00143	134	0.00187	135	-0.00067	136	-0.00056
137	0.00170	138	-0.00128	139	0.00109	140	0.00220	141	0.00104	142	-0.00048	143	-0.00005	144	0.00025
145	-0.00009	146	0.00023	147	-0.00020	148	-0.00007	149	-0.00031	150	-0.00013	151	-0.00023	152	-0.00005
153	-0.00006	154	-0.00009	155	-0.00003	156	-0.00001	157	0.00009	158	0.00006	159	0.00008	160	-0.00010

153	-0.00008	154	-0.00007	155	0.00003	156	0.00002	157	0.00014	158	0.00008	159	0.00009	160	-0.00008
TIME															
0.8334 DAYS															
72006. SECONDS															
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	-0.00000	18	0.00001	19	0.00002	20	-0.00000	21	0.00000	22	-0.00000	23	0.00001	24	0.00000
25	0.00001	26	-0.00001	27	-0.00000	28	-0.00015	29	-0.00000	30	0.00000	31	-0.00000	32	0.00002
33	0.00000	34	-0.00049	35	-0.00030	36	0.00035	37	0.00011	38	-0.00001	39	0.00109	40	-0.00000
41	0.00000	42	-0.00000	43	0.00000	44	-0.00002	45	0.00000	46	0.00106	47	-0.00085	48	0.00300
49	-0.00600	50	-0.00418	51	0.00028	52	-0.00396	53	-0.00000	54	0.00000	55	-0.00000	56	0.00001
57	-0.00002	58	0.00010	59	-0.00000	60	-0.00000	61	-0.00167	62	0.00348	63	0.00032	64	-0.02208
65	0.06094	66	-0.01267	67	0.00081	68	-0.02703	69	0.00000	70	0.00000	71	-0.00000	72	0.00000
73	-0.00003	74	0.00001	75	0.00233	76	0.00628	77	-0.00056	78	-0.02291	79	-0.02805	80	0.78607
81	0.24125	82	-0.01493	83	-0.00004	84	0.00429	85	0.00001	86	-0.00005	87	-0.00012	88	0.00050
89	-0.00065	90	-0.00232	91	0.00226	92	-0.03289	93	0.52655	94	2.79282	95	2.76726	96	2.20742
97	2.49243	98	0.49835	99	-0.01651	100	2.67376	101	0.00135	102	-0.00023	103	-0.00176	104	0.00670
105	-0.00111	106	-0.02475	107	0.41190	108	6.53478	109	7.82908	110	5.94933	111	7.80320	112	6.60143
113	2.97913	114	0.05311	115	6.59358	116	7.22725	117	1.94916	118	-0.01865	119	0.00453	120	-0.00107
121	-0.00119	122	0.00277	123	-0.03391	124	-0.01233	125	2.87391	126	0.93101	127	0.01222	128	21.96894
129	7.16439	130	-0.02337	131	-0.00803	132	0.00215	133	-0.00153	134	0.00186	135	-0.00064	136	-0.00108
137	0.00121	138	-0.00160	139	0.00066	140	0.00240	141	0.00131	142	-0.00059	143	-0.00001	144	0.00029
145	-0.00001	146	0.00031	147	-0.00008	148	0.00001	149	-0.00026	150	-0.00012	151	-0.00026	152	-0.00007
153	-0.00003	154	-0.00010	155	-0.00008	156	-0.00004	157	0.00004	158	0.00004	159	0.00006	160	-0.00013

0.8750 DAYS 75604. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00000	16	0.00000
17	0.00000	18	0.00001	19	0.00003	20	-0.00000	21	0.00000	22	-0.00000	23	0.00001	24	0.00000
25	-0.00001	26	-0.00003	27	-0.00001	28	-0.00019	29	-0.00000	30	0.00000	31	-0.00000	32	0.00003
33	0.00000	34	-0.00056	35	-0.00028	36	0.00050	37	0.00015	38	-0.00001	39	0.00133	40	-0.00000
41	0.00000	42	-0.00000	43	0.00001	44	-0.00004	45	0.00000	46	0.00153	47	-0.00081	48	0.00306
49	-0.00724	50	-0.00463	51	0.00036	52	-0.00503	53	-0.00000	54	0.00000	55	-0.00000	56	0.00002
57	-0.00003	58	0.00021	59	0.00000	60	-0.00000	61	-0.00158	62	0.00368	63	-0.00319	64	-0.02259
65	0.07103	66	-0.01098	67	0.00044	68	-0.03088	69	0.00000	70	-0.00000	71	-0.00000	72	0.00000
73	-0.00005	74	0.00007	75	0.00244	76	0.00641	77	-0.00037	78	-0.02512	79	-0.00520	80	0.85225
81	0.28267	82	-0.01398	83	-0.00051	84	0.00419	85	0.00001	86	-0.00005	87	-0.00006	88	0.00064
89	-0.00050	90	-0.00210	91	0.00200	92	-0.03460	93	0.58062	94	2.93363	95	2.90623	96	2.32606
97	2.62633	98	0.55186	99	-0.01714	100	2.79999	101	0.00111	102	-0.00007	103	-0.00163	104	0.00655
105	-0.00096	106	-0.02462	107	0.46499	108	6.70925	109	8.00646	110	6.12269	111	7.97943	112	6.77381
113	3.10874	114	0.08046	115	6.77054	116	7.40309	117	2.05798	118	-0.02102	119	0.00472	120	-0.00101
121	-0.00099	122	0.00273	123	-0.03693	124	-0.00910	125	3.01429	126	1.00626	127	0.01772	128	22.16533
129	7.33895	130	-0.02542	131	-0.01118	132	0.00249	133	-0.00158	134	0.00180	135	-0.00059	136	-0.00168
137	0.00063	138	-0.00190	139	0.00009	140	0.00253	141	0.00157	142	-0.00068	143	0.00004	144	0.00034
145	0.00008	146	0.00038	147	0.00006	148	0.00009	149	-0.00018	150	-0.00010	151	-0.00028	152	-0.00008
153	-0.00000	154	-0.00011	155	-0.00013	156	-0.00008	157	-0.00003	158	0.00001	159	0.00004	160	-0.00015

0.9167 DAYS 79203. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	0.00000	15	-0.00001	16	0.00000
17	0.00000	18	0.00002	19	0.00003	20	-0.00000	21	0.00000	22	-0.00001	23	0.00002	24	0.00000
25	-0.00003	26	-0.00006	27	-0.00002	28	-0.00023	29	0.00000	30	0.00000	31	-0.00000	32	0.00003
33	0.00000	34	-0.00062	35	-0.00024	36	0.00066	37	0.00018	38	-0.00002	39	0.00159	40	-0.00000
41	0.00000	42	-0.00000	43	0.00001	44	-0.00005	45	0.00000	46	0.00200	47	-0.00073	48	0.00302
49	-0.00864	50	-0.00508	51	0.00043	52	-0.00623	53	-0.00000	54	0.00000	55	-0.00000	56	0.00002
57	-0.00004	58	0.00033	59	0.00000	60	-0.00000	61	-0.00144	62	0.00380	63	-0.00689	64	-0.02250
65	0.08207	66	-0.00857	67	-0.00003	68	-0.03466	69	0.00000	70	-0.00000	71	-0.00000	72	0.00001
73	-0.00006	74	0.00014	75	0.00250	76	0.00637	77	-0.00015	78	-0.02693	79	0.01942	80	0.91803
81	0.32559	82	-0.01241	83	-0.00106	84	0.00398	85	0.00001	86	-0.00005	87	0.00000	88	0.00078
89	-0.00033	90	-0.00181	91	0.00167	92	-0.03568	93	0.63506	94	3.06990	95	3.04079	96	2.44126
97	2.75614	98	0.60585	99	-0.01717	100	2.92247	101	0.00082	102	0.00010	103	-0.00146	104	0.00625
105	-0.00077	106	-0.02392	107	0.51937	108	6.87629	109	8.17621	110	6.28874	111	8.14808	112	6.93890
113	3.23443	114	0.10953	115	6.93979	116	7.57137	117	2.16392	118	-0.02308	119	0.00479	120	-0.00092
121	-0.00074	122	0.00262	123	-0.03971	124	-0.00500	125	3.15008	126	1.08080	127	0.02406	128	22.35232
129	7.50603	130	-0.02700	131	-0.01446	132	0.00277	133	-0.00158	134	0.00169	135	-0.00050	136	-0.00235
137	-0.00001	138	-0.00216	139	-0.00061	140	0.00259	141	0.00183	142	-0.00076	143	0.00010	144	0.00038
145	0.00016	146	0.00043	147	0.00020	148	0.00018	149	-0.00009	150	-0.00007	151	-0.00030	152	-0.00009
153	0.00003	154	-0.00012	155	-0.00017	156	-0.00011	157	-0.00009	158	-0.00002	159	0.00001	160	-0.00018

0.9583 DAYS 82801. SECONDS

TIME

TIME									
1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000
9	0.00000	10	-0.00000	11	-0.00000	12	-0.00000	13	-0.00000
17	0.00000	18	0.00002	19	0.00004	20	-0.00000	21	0.00000
25	-0.00005	26	-0.00008	27	-0.00003	28	-0.00028	29	0.00000
33	-0.00000	34	-0.00066	35	-0.00019	36	0.00084	37	0.00020
41	0.00000	42	-0.00000	43	0.00001	44	-0.00006	45	0.00000
49	-0.01017	50	-0.00554	51	0.00049	52	-0.00752	53	-0.00000
57	-0.00004	58	0.00045	59	0.00000	60	-0.00000	61	-0.00125
65	0.09402	66	-0.00542	67	-0.00060	68	-0.03832	69	0.00000
73	-0.00008	74	0.00021	75	0.00251	76	0.00618	77	0.00010
81	0.36982	82	-0.01022	83	-0.00168	84	0.00367	85	0.00001
89	-0.00015	90	-0.00146	91	0.00127	92	-0.03611	93	0.68974
97	2.88203	98	0.66015	99	-0.01658	100	3.04137	101	0.00048
105	-0.00055	106	-0.02261	107	0.57481	108	7.03647	109	8.33890
113	3.35642	114	0.14014	115	7.10195	116	7.73266	117	2.26710
121	-0.00046	122	0.00243	123	-0.04222	124	-0.00004	125	3.28147
129	7.66623	130	-0.02807	131	-0.01782	132	0.00301	133	-0.00155
137	-0.00069	138	-0.00237	139	-0.00143	140	0.00257	141	0.00207
145	0.00023	146	0.00047	147	0.00035	148	0.00027	149	0.00002
153	0.00008	154	-0.00012	155	-0.00021	156	-0.00014	157	-0.00016
8	-0.00000	16	0.00000	24	0.00000	32	0.00004	40	-0.00000
48	0.00286	56	0.00002	64	-0.02178	72	0.00001	80	0.98333
88	0.00092	96	2.55318	104	0.00582	112	7.09726	120	-0.00080
128	22.53076	136	-0.00309	144	0.00042	152	-0.00009	160	-0.00020

TIME 1.0000 DAYS 86400. SECONDS

1	0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	0.00000	8	-0.00000
9	0.00000	10	-0.00000	11	-0.00000	12	-0.00001	13	-0.00000	14	0.00000	15	-0.00001	16	-0.00000
17	0.00001	18	0.00002	19	0.00005	20	-0.00000	21	0.00000	22	-0.00001	23	0.00002	24	0.00000
25	-0.00008	26	-0.00011	27	-0.00004	28	-0.00032	29	0.00000	30	0.00000	31	-0.00000	32	0.00004
33	-0.00000	34	-0.00069	35	-0.00012	36	0.00104	37	0.00021	38	-0.00002	39	0.00214	40	-0.00000
41	0.00000	42	-0.00000	43	0.00001	44	-0.00007	45	0.00000	46	0.00290	47	-0.00047	48	0.00260
49	-0.01182	50	-0.00598	51	0.00055	52	-0.00889	53	-0.00000	54	0.00000	55	-0.00000	56	0.00002
57	-0.00005	58	0.00057	59	0.00000	60	-0.00000	61	-0.00101	62	0.00375	63	-0.01472	64	-0.02043
65	0.10687	66	-0.00154	67	-0.00126	68	-0.04183	69	0.00000	70	-0.00000	71	-0.00000	72	0.00001
73	-0.00009	74	0.00029	75	0.00245	76	0.00585	77	0.00035	78	-0.02925	79	0.07321	80	1.04805
81	0.41514	82	-0.00740	83	-0.00238	84	0.00327	85	0.00001	86	-0.00004	87	0.00016	88	0.00104
89	0.00003	90	-0.00107	91	0.00081	92	-0.03587	93	0.74453	94	3.32962	95	3.29754	96	2.66196
97	3.00418	98	0.71462	99	-0.01534	100	3.15689	101	0.00010	102	0.00046	103	-0.00097	104	0.00526
105	-0.00029	106	-0.02070	107	0.63111	108	7.19030	109	8.49509	110	6.60106	111	8.46500	112	7.24944
113	3.47492	114	0.17212	115	7.25757	116	7.88750	117	2.36763	118	-0.02616	119	0.00456	120	-0.00064
121	-0.00015	122	0.00215	123	-0.04443	124	0.00577	125	3.40871	126	1.22740	127	0.03918	128	22.70139
129	7.82004	130	-0.02861	131	-0.02125	132	0.00319	133	-0.00147	134	0.00131	135	-0.00026	136	-0.00388
137	-0.00138	138	-0.00252	139	-0.00236	140	0.00247	141	0.00230	142	-0.00087	143	0.00022	144	0.00045
145	0.00028	146	0.00050	147	0.00051	148	0.00036	149	0.00014	150	0.00000	151	-0.00032	152	-0.00008
153	0.00012	154	-0.00012	155	-0.00024	156	-0.00017	157	-0.00023	158	-0.00008	159	-0.00005	160	-0.00021

129	10.18144	130	0.13505	131	-0.04564	132	-0.00508	133	0.00081	134	-0.00378	135	0.00025	136	0.00150
137	0.03422	138	0.04747	139	-0.01228	140	-0.01097	141	-0.00169	142	0.00056	143	-0.00053	144	-0.00111
145	-0.00032	146	0.00296	147	0.00072	148	-0.00150	149	-0.00089	150	0.00005	151	0.00033	152	-0.00038
153	-0.00047	154	-0.00011	155	0.00049	156	0.00016	157	0.00001	158	0.00001	159	0.00007	160	0.00023

TIME 2.5000 DAYS 216000. SECONDS

1	-0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00000	7	-0.00000	8	0.00000
9	0.00000	10	-0.00000	11	-0.00001	12	-0.00000	13	-0.00000	14	-0.00000	15	0.00000	16	-0.00000
17	0.00002	18	0.00004	19	0.00000	20	-0.00000	21	-0.00000	22	-0.00001	23	-0.00004	24	-0.00000
25	0.00017	26	0.00013	27	-0.00028	28	-0.00057	29	0.00000	30	0.00000	31	-0.00001	32	0.00003
33	-0.00000	34	-0.00001	35	-0.00100	36	-0.00335	37	-0.00784	38	0.00013	39	-0.00058	40	0.00000
41	0.00000	42	-0.00001	43	0.00005	44	0.00031	45	-0.00000	46	-0.00585	47	-0.00218	48	-0.02116
49	-0.03796	50	0.05563	51	-0.00133	52	0.00902	53	0.00000	54	-0.00000	55	-0.00000	56	0.00006
57	-0.00025	58	-0.00272	59	0.00000	60	0.00000	61	-0.00365	62	-0.01347	63	-0.04057	64	0.34480
65	0.90196	66	0.48270	67	-0.00080	68	0.05928	69	-0.00000	70	0.00000	71	0.00000	72	-0.00001
73	-0.00032	74	-0.00230	75	-0.00584	76	-0.00733	77	-0.00330	78	0.22229	79	1.29805	80	2.87238
81	2.02940	82	0.40343	83	0.00833	84	0.03851	85	-0.00006	86	-0.00077	87	-0.00228	88	-0.00379
89	0.00004	90	-0.00037	91	0.02727	92	0.34707	93	2.42076	94	6.16219	95	6.13264	96	5.18898
97	5.76289	98	2.38658	99	0.39163	100	5.81632	101	0.04710	102	0.00706	103	0.00081	104	0.01416
105	-0.00020	106	0.37205	107	2.48417	108	10.37547	109	11.70974	110	9.77327	111	11.67716	112	10.42710
113	6.20885	114	1.50670	115	10.46156	116	11.08363	117	4.77670	118	0.20198	119	0.00404	120	-0.00035
121	-0.00144	122	0.00547	123	0.13985	124	0.56229	125	6.23634	126	3.19497	127	0.66285	128	26.10616
129	11.00662	130	0.26872	131	-0.02184	132	-0.00653	133	-0.00037	134	0.00157	135	-0.00105	136	0.02560
137	0.08374	138	0.09968	139	0.00162	140	-0.01213	141	-0.00475	142	0.00005	143	-0.00090	144	-0.00035
145	0.00373	146	0.01001	147	0.00216	148	-0.00195	149	-0.00275	150	-0.00087	151	0.00004	152	-0.00019
153	-0.00071	154	0.00072	155	0.00068	156	0.00028	157	-0.00011	158	-0.00029	159	-0.00016	160	0.00002

TIME 3.0000 DAYS 259200. SECONDS

1	-0.00000	2	-0.00000	3	0.00000	4	-0.00000	5	0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	0.00000	10	-0.00000	11	-0.00000	12	-0.00000	13	-0.00000	14	-0.00000	15	0.00000	16	0.00000
17	0.00002	18	0.00002	19	0.00001	20	-0.00000	21	-0.00000	22	-0.00001	23	-0.00005	24	-0.00000
25	-0.00019	26	-0.00030	27	-0.00097	28	-0.00114	29	0.00000	30	-0.00000	31	-0.00001	32	0.00007
33	-0.00000	34	-0.00155	35	-0.00305	36	-0.00639	37	-0.00844	38	-0.00029	39	-0.00125	40	0.00000
41	0.00000	42	-0.00001	43	0.00006	44	-0.00009	45	-0.00000	46	-0.00993	47	-0.00529	48	-0.01742
49	-0.02030	50	0.10560	51	0.00198	52	0.04751	53	0.00000	54	-0.00000	55	0.00000	56	0.00007
57	-0.00042	58	-0.00103	59	-0.00000	60	0.00000	61	-0.00665	62	-0.00931	63	-0.00374	64	0.51294
65	1.19518	66	0.68949	67	0.02183	68	0.14917	69	-0.00000	70	0.00000	71	0.00001	72	-0.00004
73	-0.00064	74	-0.00157	75	0.00207	76	0.00671	77	-0.00252	78	0.35733	79	1.68463	80	3.35435
81	2.49002	82	0.58505	83	0.03644	84	0.07910	85	-0.00015	86	-0.00099	87	-0.00200	88	0.00121
89	0.00933	90	-0.00079	91	0.06022	92	0.53341	93	2.87724	94	6.80102	95	6.77469	96	5.77981
97	6.39135	98	2.84219	99	0.58322	100	6.43079	101	0.08976	102	0.01881	103	0.00438	104	0.04180
105	0.00042	106	0.55252	107	2.99070	108	11.06203	109	12.40009	110	10.45785	111	12.36894	112	11.11550
113	6.83907	114	1.91565	115	11.15028	116	11.77138	117	5.35102	118	0.33112	119	0.02206	120	0.00093
121	-0.00304	122	0.02453	123	0.25710	124	0.78666	125	6.87477	126	3.69871	127	0.90988	128	26.82211
129	11.69406	130	0.41977	131	0.02239	132	-0.00242	133	-0.00209	134	0.01463	135	-0.00129	136	0.06404
137	0.15127	138	0.16668	139	0.02951	140	-0.00582	141	-0.00601	142	-0.00122	143	0.00052	144	0.00359

145	0.01347	146	0.02285	147	0.00679	148	-0.00022	149	-0.00396	150	-0.00194	151	-0.00078	152	0.00182
153	0.00080	154	0.00359	155	0.00130	156	0.00063	157	-0.00084	158	-0.00098	159	-0.00075	160	-0.00068

302400. SECONDS

3.5000 DAYS

TIME

1	0.00000	2	-0.00000	3	0.00000	4	0.00000	5	0.00000	6	-0.00000	7	-0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00002	12	0.00001	13	-0.00000	14	-0.00000	15	0.00001	16	0.00000
17	-0.00005	18	-0.00011	19	-0.00004	20	-0.00000	21	-0.00000	22	-0.00001	23	-0.00011	24	0.00000
25	-0.00090	26	-0.00106	27	-0.00179	28	-0.00183	29	0.00000	30	-0.00000	31	-0.00000	32	0.00003
33	0.00000	34	-0.00338	35	-0.00485	36	-0.00815	37	-0.00594	38	-0.00056	39	0.00078	40	0.00000
41	-0.00000	42	0.00000	43	0.00003	44	-0.00032	45	0.00000	46	-0.01117	47	-0.00727	48	-0.00532
49	0.01040	50	0.16751	51	0.00927	52	0.09982	53	0.00000	54	-0.00000	55	0.00001	56	0.00002
57	-0.00009	58	0.00441	59	-0.00000	60	-0.00000	61	-0.00715	62	0.00269	63	0.05017	64	0.68816
65	1.48353	66	0.90081	67	0.05613	68	0.25535	69	-0.00000	70	0.00001	71	0.00002	72	-0.00008
73	-0.00052	74	0.00230	75	0.01849	76	0.03189	77	0.00306	78	0.50338	79	2.04868	80	3.79069
81	2.91174	82	0.77311	83	0.07728	84	0.13323	85	-0.00020	86	-0.00013	87	0.00144	88	0.01354
89	0.02782	90	0.00276	91	0.10459	92	0.72893	93	3.29255	94	7.35457	95	7.33061	96	6.29712
97	6.93678	98	3.25687	99	0.78308	100	6.96617	101	0.14431	102	0.03908	103	0.01362	104	0.08165
105	0.00498	106	0.74051	107	3.44814	108	11.64958	109	12.99045	110	11.04399	111	12.96051	112	11.70487
113	7.38681	114	2.29707	115	11.73922	116	12.35966	117	5.85606	118	0.47223	119	0.05063	120	0.00593
121	-0.00155	122	0.05382	123	0.38825	124	1.01255	125	7.42761	126	4.15038	127	1.15792	128	27.43077
129	12.28231	130	0.58043	131	0.08300	132	0.00837	133	-0.00284	134	0.03582	135	0.00140	136	0.11504
137	0.23319	138	0.24540	139	0.07012	140	0.00869	141	-0.00418	142	-0.00240	143	0.00520	144	0.01200
145	0.02995	146	0.04203	147	0.01574	148	0.00468	149	-0.00341	150	-0.00250	151	-0.00174	152	0.00705
153	0.00548	154	0.00972	155	0.00326	156	0.00177	157	-0.00199	158	-0.00178	159	-0.00147	160	-0.00155

345600. SECONDS

4.0000 DAYS

TIME

1	0.00000	2	-0.00000	3	0.00000	4	0.00000	5	0.00000	6	-0.00001	7	0.00000	8	0.00000
9	-0.00000	10	-0.00000	11	0.00005	12	0.00003	13	-0.00000	14	-0.00000	15	0.00002	16	0.00000
17	-0.00019	18	-0.00034	19	-0.00015	20	-0.00000	21	-0.00000	22	0.00000	23	-0.00021	24	0.00000
25	-0.00173	26	-0.00189	27	-0.00244	28	-0.00230	29	-0.00000	30	-0.00000	31	0.00001	32	-0.00010
33	0.00000	34	-0.00467	35	-0.00555	36	-0.00768	37	0.00029	38	-0.00027	39	0.00637	40	-0.00000
41	-0.00000	42	0.00002	43	-0.00002	44	0.00010	45	0.00000	46	-0.00834	47	-0.00667	48	0.01491
49	0.05242	50	0.23905	51	0.02108	52	0.16343	53	-0.00000	54	-0.00000	55	0.00001	56	-0.00007
57	0.00123	58	0.01433	59	-0.00000	60	-0.00000	61	-0.00336	62	0.02271	63	0.11739	64	0.86582
65	1.76285	66	1.11176	67	0.10085	68	0.37279	69	0.00000	70	0.00001	71	0.00000	72	-0.00005
73	0.00060	74	0.01033	75	0.04380	76	0.06750	77	0.01489	78	0.65551	79	2.39014	80	4.18866
81	3.29891	82	0.96307	83	0.12973	84	0.19918	85	-0.00003	86	0.00254	87	0.00928	88	0.03428
89	0.05653	90	0.01213	91	0.15895	92	0.92750	93	3.67257	94	7.84292	95	7.82047	96	6.75720
97	7.41840	98	3.63645	99	0.98532	100	7.44035	101	0.20906	102	0.06877	103	0.03013	104	0.13217
105	0.01514	106	0.93082	107	3.86344	108	12.16304	109	13.50608	110	11.55639	111	13.47707	112	12.21985
113	7.87066	114	2.65206	115	12.25359	116	12.87354	117	6.30655	118	0.62018	119	0.08896	120	0.01609
121	0.00484	122	0.09259	123	0.52791	124	1.23564	125	7.91493	126	4.55926	127	1.40211	128	27.95998
129	12.79625	130	0.74564	131	0.15624	132	0.02616	133	-0.00116	134	0.06498	135	0.00860	136	0.17653
137	0.32606	138	0.33302	139	0.12159	140	0.03118	141	0.00150	142	-0.00243	143	0.01436	144	0.02581
145	0.05357	146	0.06760	147	0.02962	148	0.01336	149	-0.00030	150	-0.00189	151	-0.00222	152	0.01669
153	0.01459	154	0.02009	155	0.00755	156	0.00439	157	-0.00307	158	-0.00221	159	-0.00189	160	-0.00206

TIME		4.5000 DAYS				388800. SECONDS									
1	0.00000	2	-0.00000	3	0.00000	4	-0.00000	5	0.00000	6	-0.00001	7	0.00001	8	-0.00000
9	-0.00000	10	-0.00000	11	0.00009	12	0.00006	13	-0.00000	14	-0.00000	15	0.00003	16	0.00000
17	-0.00038	18	-0.00062	19	-0.00033	20	-0.00000	21	0.00000	22	0.00001	23	-0.00033	24	0.00000
25	-0.00238	26	-0.00242	27	-0.00257	28	-0.00218	29	-0.00000	30	0.00000	31	0.00002	32	-0.00029
33	0.00000	34	-0.00461	35	-0.00441	36	-0.00434	37	0.01054	38	0.00097	39	0.01601	40	-0.00000
41	-0.00000	42	0.00002	43	-0.00005	44	0.00162	45	0.00000	46	-0.00072	47	-0.00239	48	0.04262
49	0.10390	50	0.31814	51	0.03758	52	0.23600	53	-0.00000	54	0.00001	55	-0.00001	56	-0.00015
57	0.00398	58	0.02907	59	-0.00000	60	-0.00000	61	0.00591	62	0.05049	63	0.19476	64	1.04302
65	2.03117	66	1.31951	67	0.15456	68	0.49781	69	0.00001	70	-0.00001	71	-0.00006	72	0.00012
73	0.00326	74	0.02314	75	0.07781	76	0.11249	77	0.03376	78	0.81042	79	2.71014	80	4.55402
81	3.65588	82	1.15222	83	0.19245	84	0.27519	85	0.00057	86	0.00769	87	0.02236	88	0.06374
89	0.09562	90	0.02848	91	0.22185	92	1.12533	93	4.02231	94	8.27988	95	8.25824	96	7.17151
97	7.84959	98	3.98594	99	1.18635	100	7.86590	101	0.28242	102	0.10806	103	0.05476	104	0.19178
105	0.03197	106	1.12023	107	4.24281	108	12.61909	109	13.96389	110	12.01165	111	13.93552	112	12.67710
113	8.30380	114	2.98272	115	12.71020	116	13.32983	117	6.71305	118	0.77147	119	0.13598	120	0.03238
121	0.01734	122	0.13986	123	0.67234	124	1.45355	125	8.35060	126	4.93247	127	1.63975	128	28.42815
129	13.25260	130	0.91213	131	0.23894	132	0.05078	133	0.00417	134	0.10169	135	0.02143	136	0.24657
137	0.42699	138	0.42720	139	0.18199	140	0.06097	141	0.01144	142	-0.00029	143	0.02892	144	0.04561
145	0.08431	146	0.09938	147	0.04866	148	0.02611	149	0.00593	150	0.00050	151	-0.00149	152	0.03162
153	0.02902	154	0.03541	155	0.01505	156	0.00917	157	-0.00346	158	-0.00169	159	-0.00148	160	-0.00156

TIME		5.0000 DAYS		432000. SECONDS																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
------	--	-------------	--	-----------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

5.5000 DAYS 475200. SECONDS

TIME

1	-0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	0.00000	6	0.00001	7	0.00002	8	-0.00000
9	0.00000	10	-0.00000	11	0.00010	12	0.00008	13	-0.00000	14	0.00000	15	0.00002	16	-0.00000
17	-0.00062	18	-0.00084	19	-0.00067	20	0.00000	21	-0.00000	22	-0.00001	23	-0.00042	24	-0.00000
25	-0.00170	26	-0.00104	27	-0.00000	28	0.00126	29	0.00000	30	0.00000	31	-0.00003	32	-0.00060
33	-0.00000	34	0.00222	35	0.00547	36	0.01226	37	0.04311	38	0.00763	39	0.04785	40	-0.00000
41	0.00001	42	-0.00005	43	0.00026	44	0.00948	45	-0.00000	46	0.03005	47	0.01977	48	0.11733
49	0.22847	50	0.49209	51	0.08431	52	0.40040	53	-0.00000	54	0.00006	55	-0.00018	56	0.00011
57	0.01524	58	0.07313	59	-0.00000	60	0.00000	61	0.04317	62	0.12720	63	0.37037	64	1.38959
65	2.53285	66	1.71989	67	0.28356	68	0.76075	69	0.00003	70	-0.00014	71	-0.00039	72	0.00136
73	0.01499	74	0.06417	75	0.16979	76	0.22616	77	0.09340	78	1.12065	79	3.29248	80	5.20463
81	4.29429	82	1.52228	83	0.34336	84	0.45119	85	0.00393	86	0.02717	87	0.06563	88	0.14786
89	0.20298	90	0.08390	91	0.36823	92	1.51072	93	4.64684	94	9.03648	95	9.01501	96	7.89396
97	8.59666	98	4.61051	99	1.57743	100	8.60521	101	0.44956	102	0.21398	103	0.12905	104	0.33275
105	0.08746	106	1.48949	107	4.91385	108	13.40254	109	14.75004	110	12.79394	111	14.72227	112	13.46206
113	9.05378	114	3.58032	115	13.49408	116	14.11334	117	7.42337	118	1.07559	119	0.25178	120	0.08498
121	0.06314	122	0.25605	123	0.96606	124	1.86966	125	9.10410	126	5.59292	127	2.09071	128	29.22887
129	14.03613	130	1.24125	131	0.42305	132	0.11881	133	0.02894	134	0.19556	135	0.06649	136	0.40587
137	0.64426	138	0.62808	139	0.32263	140	0.13907	141	0.04456	142	0.01368	143	0.07636	144	0.10404
145	0.16593	146	0.18006	147	0.10201	148	0.06408	149	0.02875	150	0.01256	151	0.00628	152	0.07927
153	0.07600	154	0.08235	155	0.04218	156	0.02742	157	0.00042	158	0.00438	159	0.00387	160	0.00501

6.0000 DAYS 518400. SECONDS

TIME

1	-0.00000	2	0.00000	3	-0.00000	4	-0.00000	5	-0.00000	6	0.00002	7	0.00003	8	0.00000
9	0.00000	10	-0.00000	11	0.00004	12	0.00006	13	-0.00000	14	-0.00000	15	-0.00002	16	-0.00000
17	-0.00050	18	-0.00050	19	-0.00069	20	0.00000	21	-0.00001	22	-0.00003	23	-0.00025	24	-0.00000
25	0.00027	26	0.00155	27	0.00323	28	0.00511	29	0.00000	30	-0.00000	31	-0.00010	32	-0.00054
33	-0.00000	34	0.00989	35	0.01487	36	0.02577	37	0.06514	38	0.01351	39	0.06983	40	-0.00001
41	0.00001	42	-0.00016	43	0.00077	44	0.01632	45	-0.00000	46	0.05320	47	0.03822	48	0.16275
49	0.29873	50	0.58431	51	0.11406	52	0.48929	53	-0.00000	54	0.00010	55	-0.00038	56	0.00066
57	0.02421	58	0.10213	59	0.00000	60	0.00000	61	0.07139	62	0.17489	63	0.46513	64	1.55731
65	2.76664	66	1.91140	67	0.35655	68	0.89546	69	0.00004	70	-0.00028	71	-0.00070	72	0.00266
73	0.02467	74	0.09234	75	0.22631	76	0.29278	77	0.13381	78	1.27366	79	3.55822	80	5.49675
81	4.58194	82	1.70183	83	0.42910	84	0.54857	85	0.00706	86	0.04200	87	0.09586	88	0.20138
89	0.26961	90	0.12288	91	0.44960	92	1.69634	93	4.92798	94	9.36889	95	9.34705	96	8.21315
97	8.92509	98	4.89192	99	1.76579	100	8.93093	101	0.54113	102	0.27925	103	0.17817	104	0.41186
105	0.12613	106	1.66772	107	5.21344	108	13.74470	109	15.09323	110	13.13564	111	15.06555	112	13.80463
113	9.38323	114	3.85158	115	13.83622	116	14.45542	117	7.73770	118	1.22594	119	0.31865	120	0.12133
121	0.09675	122	0.32323	123	1.11253	124	2.06720	125	9.43483	126	5.8805	127	2.30336	128	29.57736
129	14.37813	130	1.40173	131	0.52097	132	0.16118	133	0.04925	134	0.25160	135	0.09914	136	0.49267
137	0.75745	138	0.73220	139	0.40003	140	0.18570	141	0.06764	142	0.02662	143	0.10963	144	0.14261
145	0.21589	146	0.22811	147	0.13587	148	0.08912	149	0.04551	150	0.02294	151	0.01445	152	0.11230
153	0.10894	154	0.11415	155	0.06255	156	0.04171	157	0.00581	158	0.01085	159	0.00971	160	0.01216

6.5000 DAYS 561600. SECONDS

TIME

1	-0.00000	2	-0.00000	3	-0.00000	4	0.00000	5	-0.00000	6	0.00003	7	0.00002	8	0.00000
9	0.00001	10	-0.00000	11	-0.00008	12	-0.00000	13	-0.00000	14	-0.00001	15	-0.00007	16	-0.00000
17	-0.00009	18	0.00033	19	-0.00049	20	-0.00000	21	-0.00003	22	-0.00006	23	0.00016	24	-0.00000
25	0.00368	26	0.00576	27	0.00804	28	0.01064	29	0.00000	30	-0.00001	31	-0.00021	32	-0.00022
33	-0.00001	34	0.02074	35	0.02747	36	0.04271	37	0.09068	38	0.02127	39	0.09555	40	-0.00001
41	0.00001	42	-0.00034	43	0.00161	44	0.02530	45	-0.00000	46	0.08128	47	0.06159	48	0.21261
49	0.37282	50	0.67870	51	0.14765	52	0.58118	53	0.00000	54	0.00015	55	-0.00066	56	0.00160
57	0.03558	58	0.13542	59	0.00000	60	0.00000	61	0.10575	62	0.22795	63	0.56287	64	1.72083
65	2.98983	66	2.09699	67	0.43392	68	1.03093	69	0.00006	70	-0.00046	71	-0.00111	72	0.00451
73	0.03710	74	0.12535	75	0.28885	76	0.36472	77	0.18075	78	1.42442	79	3.80914	80	5.77038
81	4.85192	82	1.87738	83	0.52028	84	0.65076	85	0.01134	86	0.06032	87	0.13153	88	0.26157
89	0.34364	90	0.16892	91	0.53528	92	1.87679	93	5.19174	94	9.67681	95	9.65444	96	8.50970
97	9.22947	98	5.15611	99	1.94900	100	9.23314	101	0.63687	102	0.35167	103	0.23458	104	0.49554
105	0.17172	106	1.84129	107	5.49323	108	14.06075	109	15.41016	110	13.45125	111	15.38244	112	14.12091
113	9.68840	114	4.10704	115	14.15214	116	14.77132	117	8.02995	118	1.37421	119	0.39041	120	0.16408
121	0.13732	122	0.39549	123	1.25766	124	2.25783	125	9.74105	126	6.16384	127	2.50761	128	29.89868
129	14.69392	130	1.55871	131	0.62114	132	0.20845	133	0.07504	134	0.31301	135	0.13845	136	0.58299
137	0.87222	138	0.83756	139	0.48067	140	0.23640	141	0.09489	142	0.04395	143	0.14923	144	0.18716
145	0.27129	146	0.28071	147	0.17413	148	0.11800	149	0.06580	150	0.03654	151	0.02603	152	0.15134
153	0.14813	154	0.15139	155	0.08766	156	0.05976	157	0.01406	158	0.02006	159	0.01813	160	0.02244

TIME 7.0000 DAYS 604800. SECONDS

1	-0.00000	2	-0.00000	3	-0.00000	4	0.00000	5	-0.00000	6	0.00005	7	0.00001	8	0.00000
9	0.00000	10	-0.00000	11	-0.00027	12	-0.00011	13	-0.00000	14	-0.00002	15	-0.00015	16	-0.00000
17	0.00068	18	0.00177	19	0.00000	20	-0.00000	21	-0.00007	22	-0.00010	23	0.00088	24	0.00000
25	0.00873	26	0.01178	27	0.01454	28	0.01797	29	0.00000	30	-0.00005	31	-0.00035	32	0.00043
33	-0.00001	34	0.03487	35	0.04328	36	0.06298	37	0.11947	38	0.03098	39	0.12472	40	-0.00001
41	0.00000	42	-0.00059	43	0.00284	44	0.03648	45	-0.00000	46	0.11399	47	0.08973	48	0.26629
49	0.44987	50	0.77455	51	0.18475	52	0.67526	53	0.00001	54	0.00021	55	-0.00104	56	0.00302
57	0.04940	58	0.17267	59	0.00000	60	0.00001	61	0.14590	62	0.28580	63	0.66270	64	1.88005
65	3.20312	66	2.27678	67	0.51490	68	1.16651	69	0.00007	70	-0.00071	71	-0.00164	72	0.00700
73	0.05235	74	0.16289	75	0.35666	76	0.44123	77	0.23368	78	1.57265	79	4.04669	80	6.02769
81	5.10631	82	2.04891	83	0.61600	84	0.75690	85	0.01684	86	0.08208	87	0.17230	88	0.32770
89	0.42413	90	0.22153	91	0.62461	92	2.05206	93	5.44015	94	9.96366	95	9.94065	96	8.78664
97	9.51317	98	5.40510	99	2.12708	100	9.51509	101	0.73607	102	0.43040	103	0.29766	104	0.58310
105	0.22382	106	2.01021	107	5.75566	108	14.35448	109	15.70463	110	13.74456	111	15.67680	112	14.41471
113	9.97266	114	4.34833	115	14.44563	116	15.06485	117	8.30307	118	1.52007	119	0.46641	120	0.21286
121	0.18450	122	0.47224	123	1.40106	124	2.44180	125	10.02621	126	6.42266	127	2.70381	128	30.19682
129	14.98729	130	1.71196	131	0.72271	132	0.26015	133	0.10625	134	0.37932	135	0.18416	136	0.67615
137	0.98784	138	0.94355	139	0.56371	140	0.29059	141	0.12614	142	0.06581	143	0.19494	144	0.23737
145	0.33163	146	0.33745	147	0.21648	148	0.15050	149	0.08953	150	0.05348	151	0.04128	152	0.19618
153	0.19335	154	0.19383	155	0.11750	156	0.08169	157	0.02548	158	0.03226	159	0.02938	160	0.03612

FINAL BANDWIDTH

21

B30420